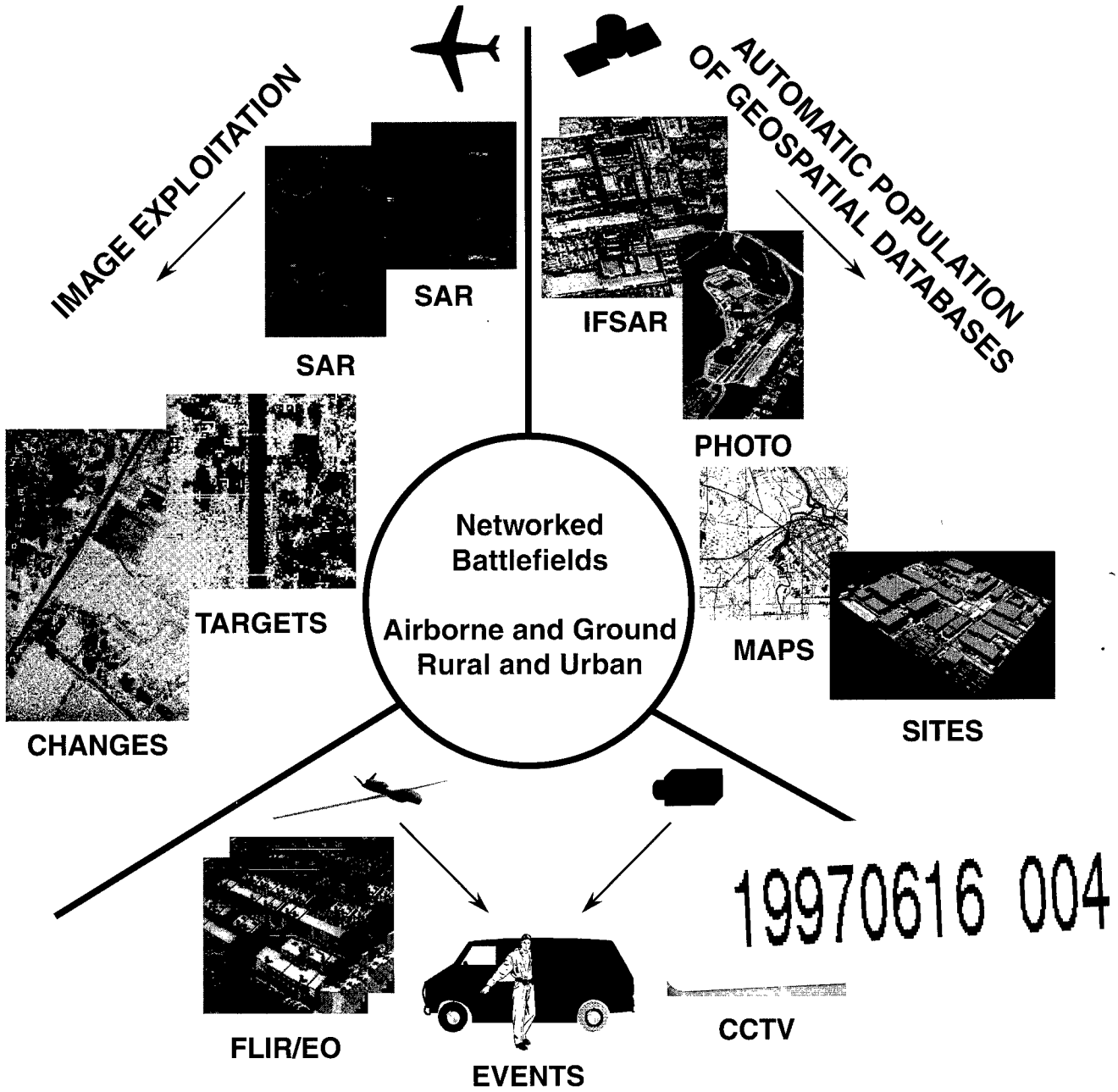
DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

Proceedings
1997 Image Understanding Workshop
11 - 14 May
Hyatt Regency, New Orleans, LA

Volume I



VIDEO SURVEILLANCE AND MONITORING

19970616 004

Edited by
Thomas M. Strat

Sponsored by
Defense Advanced Research Projects Agency
Information Systems Office

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
Proceedings, 1997 Image Understanding Workshop, 11-14 May, Hyatt Regency, New Orleans, LA

5. FUNDING NUMBERS

6. AUTHOR(S)
Thomas M. Strat, Image Understanding Program Manager

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

DARPA Information Systems Office
3701 N. Fairfax Drive
Arlington, VA 22203

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

DARPA Information Systems Office
3701 N. Fairfax Drive
Arlington, VA 22203

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

This proceedings contains the assembled reports of the various research projects that comprise the DARPA Image Understanding (IU) Program. Submissions from forty academic and industrial computer vision research laboratories document progress and lessons learned in research performed for applications in image registration, target recognition, image exploitation, cartography, 3D model reconstruction, video surveillance, activity recognition and content-based image retrieval.

This is the 25th proceedings in the series, which has become known as a comprehensive source for the latest research results in image understanding from the nation's leading IU laboratories. Like its predecessors, this proceedings is not peer-reviewed in the traditional sense of a refereed conference or journal. Instead, the principal investigator of each laboratory is responsible for selecting the papers that will represent the work carried out in his lab. Because the reputation of each lab is at stake, the quality of submissions has remained consistently high.

14. SUBJECT TERMS

15. NUMBER OF PAGES
1531

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclass	Unclass	Unclass	

IMAGE UNDERSTANDING WORKSHOP

Proceedings of a Workshop

held in

New Orleans, Louisiana

May 11 - 14, 1997

Volume I

Sponsored by:

Defense Advanced Research Projects Agency
Information Systems Office

This document contains copies of reports prepared for the DARPA Image Understanding Workshop. Included are Principal Investigator reports and technical results from both the basic and strategic computing programs within DARPA/ISO sponsored projects and certain technical reports from selected scientists from other organizations.

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Government of the United States of America.

DTIC QUALITY INSPECTED 2

Distributed by:
Morgan Kaufmann Publishers, Inc.
340 Pine Street 6th Floor
San Francisco, CA 94104-3205
ISBN: 1-55860-490-1
Printed in the United States of America

Printed by: ***Omnipress*** *. . . .* *"Helping Associations Educate"*
Madison, Wisconsin (800) 828-0305

TABLE OF CONTENTS

Table of Contents iii

Author Index xi

Foreword xv

Acknowledgements xix

VOLUME I

SECTION I: VIDEO SURVEILLANCE AND MONITORING (VSAM)

Video Surveillance and Monitoring - Principal Investigator Reports

"Cooperative Multi-Sensor Video Surveillance," Takeo Kanade, Robert T. Collins, Alan J. Lipton, P. Anandan, Peter Burt, and Lambert Wixson 3

"Video Processing for Security, Surveillance and Monitoring," P. Anandan and Peter Burt 11

"Visual Surveillance and Monitoring of Human and Vehicular Activity," Larry Davis, Rama Chellappa, Yaser Yacoob, and Qinfen Zheng 19

"VSAM at the MIT Media Laboratory and CBCL: Learning and Understanding Action in Video Imagery," Aaron Bobick, Alex Pentland, and Tommy Poggio 25

"Surveillance and Monitoring Using Video Images from a UAV," Gerard Medioni and Ram Nevatia 31

"Extra Set of Eyes," Robert C. Bolles, Kurt G. Konolige, and Martin A. Fischler 41

"A Forest of Sensors," E. Grimson, P. Viola, O. Faugeras, T. Lozano-Perez, T. Poggio, and S. Teller 45

"Robust Video Motion Detection and Event Recognition," Bruce Flinchbaugh 51

"Omnidirectional VSAM Systems: PI Report," Shree K. Nayar and Terrance E. Boult 55

"Image-Based Scene Rendering and Manipulation Research at the University of Wisconsin," Charles R. Dyer 63

"Image Understanding Research at Rochester," Christopher Brown and Randal Nelson 69

"Multi-Sensor Representation of Extended Scenes using Multi-View Geometry," Shmuel Peleg, Amnon Shashua, Daphna Weinshall, Michael Werman and Michal Irani 79

"Project Plan for Multiple Perspective Interactive Video Surveillance and Monitoring," Ramesh Jain 85

"Image Understanding at Cornell University," Daniel P. Huttenlocher and Ramin Zabih 89

"Image Understanding Research at CMU," T. Kanade and K. Ikeuchi 95

Video Surveillance and Monitoring - Technical Papers

"Exploring Visual Motion Using Projections of Motion Fields," Sandor Fejes and Larry S. Davis 113

"A Kalman Filter That Learns Robust Models of Dynamic Phenomena," Rajesh P.N. Rao 123

"Learning to Fixate on 3D Targets With Uncalibrated Active Cameras," Narayan Srinivasa and Narendra Ahuja 129

"Temporal Multi-scale Models for Image Motion Estimation," Yaser Yacoob and Larry S. Davis 135

"Understanding Object Motion," Zoran Duric, Ehud Rivlin, and Azriel Rosenfeld 143

"Representing Local Motion as a Probability Distribution Matrix and Object Tracking," Yoav Rosenberg and Michael Werman 153

"Moving Object Detection and Event Recognition Algorithms for Smart Cameras," Thomas J. Olson and Frank Z. Brill 159

"Sensory Attention: Computational Sensor Paradigm for Low-Latency Adaptive Vision," Vladimir Brajovic and Takeo Kanade 177

"Experiments with an Algorithm for Recovering Fluid Flow from Video Imagery." R.P. Wildes, M.J. Amabile, A.M. Lanzillotto, and T.S. Leu 185

"Real-Time 3-D Tracking and Classification of Human Behavior," Alex Pentland, Ali Azarbayjani, Nuria Oliver, and Matt Brand 193

"Modeling and Prediction of Human Behavior," Alex Pentland and Andrew Liu 201

"A Trainable System for People Detection," Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, and Tomaso Poggio 207

"Self-Taught Visually-Guided Pointing for a Humanoid Robot," M. Marjanovic, B. Scassellati, and M. Williamson 215

"Performance and Human Interface Issues of a System for Visual Interpretation of Hand Gestures," Rick Kjeldsen and John R. Kender 221

"PNF Propagation and the Detection of Actions Described by Temporal Intervals," Claudio Pinhanez and Aaron Bobick 227

"Omnidirectional Video Camera," Shree K. Nayar 235

"Generation of Perspective and Panoramic Video from Omnidirectional Video," Venkata N. Peri and Shree K. Nayar 243

"An Integrated Approach to Image Stabilization, Mosaicking and Super-resolution," S. Srinivasan and R. Chellappa 247

"Mosaicking with Generalized Strips," Benny Rousso, Shmuel Peleg, and Ilan Finci 255

"Panoramic Mosaics with VideoBrush," Shmuel Peleg and Joshua Herman 261

"Multi-Image Alignment," Harpreet S. Sawhney and Rakesh Kumar 265

"Horizon Line Matching for Orientation Correction Using a Messy Genetic Algorithm," Karthik Balasubramaniam, J. Ross Beveridge, Christopher E. Lesher, and Christopher Graves 275

"Fast Image Stabilization and Mosaicking," Carlos Morimoto, Rama Chellappa, and Stephen Balakirsky ... 285

"Evaluation of Image Stabilization Algorithms," Carlos Morimoto and Rama Chellappa 295

"Multiple Perspective Interactive Video Surveillance and Monitoring," Jeffrey Boyd, Edward Hunter, Ramesh Jain, Patrick Kelly, Jennifer Schlenzig, and Andy Tai 303

"Sketch-First Modeling of Buildings from Video Imagery," Bob Bolles, Marty Fischler, Marsha Jo Hannah, Tuan Luong, Riadh Munjy, and Mushtaq Hussain 325

"Visibility Estimation from a Moving Vehicle Using the RALPH Vision System," Dean Pomerleau 339

"A Rapidly Adapting Machine Vision System for Automated Vehicle Steering," Dean Pomerleau and Todd Jochem 345

"Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques," Michael A. Smith and Takeo Kanade 357

"Mixed Traffic and Automated Highways," Chuck Thorpe 367

"Optic Flow Estimation from 3D Wavelet Edge Detection," Andrew Lundberg and Lawrence B. Wolf 375

SECTION II: IMAGE EXPLOITATION (IMEX)

Image Exploitation - Principal Investigator Reports

"The RADIUS Phase II Program," Anthony Hoogs, Bill Bremner, and Doug Hackett 381

"RADIUS Technology Transfer," Donald J. Gerson, Sidney E. Wood, and William R. Glatz 401

"FOCUS: A Shared Vision Technology Transfer Project," Eamon B. Barrett, Paul M. Payton, and Joseph L. Mundy 407

"Temporal Analysis of Vehicular Activities from SAR/EO," R. Chellappa, P. Burlina, Q. Zheng, C. Shekhar, and L.S. Davis 415

"Image Understanding Research at GE," J.L. Mundy 425

"Continouous Terrain Modeling from Image Sequences with Applications to Change Detection," Yvan G. Leclerc 431

"USC RADIUS Related Research: An Overview," R. Nevatia and A. Huertas 437

"A Real-Time, Interactive SAR Tactical Mapper," John B. Hampshire II 449

"Image Understanding at Lockheed Martin Valley Forge," Anthony Hoogs, Doug Hackett, and Tom Barrett 455

"IU at UI: An Overview of Research During 1996-97," Narendra Ahuja and Thomas Huang 465

"Image Understanding Research at Hughes Aircraft Company: Adaptive Image Exploitation," David M. Doria 475

"Image Understanding Research at UC Riverside: Integrated Recognition, Learning and Image Databases," Bir Bhanu 483

"Image Browsing and Retrieval Research at Stanford," Leonidas J. Guibas and Carlo Tomasi 495

Image Exploitation - Technical Papers

"Thermal Invariants for Material Labeling and Site Monitoring Using Midwave Infrared Imagery: Initial Results," C. Stewart, V. Snell, D. Hamilton, and J. Mundy 503

"Texture Segmentation of SAR Images," By-Her Wang and Thomas O. Binford 513

"Sketching Natural Terrain from Uncalibrated Imagery," Q.-T. Luong 519

"Design of Self-Tuning IU Systems," Chandra Shekhar, Philippe Burlina, and Sabine Moisan 529

"Super Resolution with Region Sensitive Interpolation," Krishna Ratakonda and Narendra Ahuja 537

"Hierarchical Image Segmentation Using Similarity Analysis," Peter Bajcsy and Narendra Ahuja 541

"sarMapper: A Real-Time, Interactive SAR Tactical Mapper," John B. Hampshire II 547

"Combining Geometric and Appearance Models for Change Detection," Anthony Hoogs 565

"Sensitivity Analysis and Learning Strategies for Context-Based Vehicle Detection Algorithms," P. Burlina, V. Parameswaran, and R. Chellappa 577

"Using RADIUS Site Models without the RCDE," Aaron J. Heller, Christopher I. Connolly, and Yvan G. Leclerc 585

"Grouping Planar Projective Symmetries," R.W. Curwen and J. L. Mundy 595

"User Interface Representations for Image Understanding," Michael A.J. Puscar and Anthony Hoogs 607

"A Geometric Framework for Image Alignment," Venu Govindu, Chandra Shekhar, and Rama Chellappa .. 615

"Multiple View 2D-3D Mutual Information Registration," M.E. Leventon, W.M. Wells III, and W.E.L. Grimson 625

"Minimizing Algebraic Error," Richard I. Hartley 631

"Robust Multi-Sensor Image Alignment," Michal Irani and P. Anandan 639

"The Cubic Rational Polynomial Camera Model," Richard I. Hartley and Tushar Saxena 649

"Rosetta: An Image Database Retrieval System," Jeremy S. De Bonet and Paul Viola 655

"The Earth Mover's Distance, Multi-Dimensional Scaling, and Color-Based Image Retrieval," Yossi Rubner, Leonidas Guibas, and Carlo Tomasi 661

"Shape-based Image Retrieval Using Geometric Hashing," Scott D. Cohen and Leonidas J. Guibas 669

"Configuration Based Scene Classification and Image Indexing," Pamela R. Lipson, Eric Grimson, and Pawan Sinha 675

"Extracting Templates for Scene Classification using a Few Examples," A. Lakshmi Ratan and W.E.L. Grimson 681

"Combining Color and Spatial Information for Content-based Image Retrieval," Jing Huang and Ramin Zabih 687

"A Characterization of Visual Appearance Applied to Image Retrieval," S. Ravela and R. Manmatha 693

"Feature Selection for Robust Color Image Retrieval," Madirakshi Das and Edward M. Riseman 701

"Automatic Text Detection and Recognition," Victor Wu, R. Manmatha, and Edward M. Riseman 707

Section III: Image Understanding Environment (IUE)

Image Understanding Environment Technical Papers

"The Image Understanding Environment Progress since IUW'96," Richard A. Lerner 715

"Programming in the Image Understanding Environment: Locating Fibers in Microscope Images," Richard A. Lerner and John Dolan 723

Volume II

Section IV: Automatic Population of Geospatial Databases (APGD)

Automatic Population of Geospatial Databases - Principal Investigator Reports

"An Integrated Feasability Demonstration for Automatic Population of Geospatial Databases," Martin A. Fischler, Robert C. Bolles, and Aaron J. Heller 759

"Automatic Acquisition of Hierarchical, Textured 3D Geometric Models of Urban Environments: Project Plan," Seth Teller 767

"Knowledge-Based Automatic Feature Extraction," Ram Nevatia and Keith Price 771

"Research in the Automated Analysis of Remotely Sensed Imagery: 1995-1996," David M. McKeown, Jr., Michael Bowling, G. Edward Bulwinkle, Steven Douglas Cochran, Stephen J. Ford, Wilson A. Harvey, Dirk Kalp, Jeff McMahill, Chris McGlone, Michael F. Polis, Jefferey A. Shufelt, and Daniel Yocum 779

"Progress in Computer Vision at the University of Massachusetts," Allen R. Hanson, Edward M. Riseman, and Howard Schultz 813

"IU at the University of Utah: Extraction of Micro-Terrain Features," William B. Thompson and Thomas C. Henderson 819

"Image Understanding Research at Colorado State University," Bruce A. Draper and J. Ross Beveridge 825

"Learning to Detect Rooftops in Aerial Images," Marcus A. Maloof, Pat Langley, Stephanie Sage, and Thomas O. Binford 835

Automatic Population of Geospatial Databases - Technical Papers

"Geospatial Registration," Rakesh Kumar, Harpreet S. Sawhney, and Jane C. Asmuth 849

"Matching and Pose Refinement with Camera Pose Estimates," Satyan Coorg and Seth Teller 857

"Unifying Two-View and Three-View Geometry," Shai Avidan and Amnon Shashua 863

"Multi-Image Correspondence using Geometric and Structural Constraints," George T. Chou and Seth Teller 869

"A Full-Projective Improvement for Lowe's Pose-estimation Algorithm," Helder Araujo, Rodrigo L. Carceroni, and Chris Brown 875

"View Morphing: Uniquely Predicting Scene Appearance from Basis Images," Steven M. Seitz and Charles R. Dyer 881

"Direct Methods for Estimation of Structure and Motion from Three Views," G.P. Stein and A. Shashua 889

"Dense Depth Maps from Epipolar Images," J.P. Mellor, Seth Teller, and Tomas Lozano-Perez 893

"Geometric Constraint Analysis and Synthesis: Methods for Improving Shape-Based Registration Accuracy," David A. Simon and Takeo Kanade 901

"Consensus Surfaces for Modeling 3D Objects from Multiple Range Images," Mark D. Wheeler, Yoichi Sato, and Katsushi Ikeuchi 911

"Solid Model Construction using Meshes and Volumes," Michael K. Reed and Peter K. Allen 921

"Constraint Optimization and Feature-Based Model Construction for Reverse Engineering," H. James de St. Germain, Stevan R. Stark, William B. Thompson, and Thomas C. Henderson 927

"Photorealistic Scene Reconstruction by Voxel Coloring," Steven M. Seitz and Charles R. Dyer 935

"Edge-aligning Surface Fitting Using Triangular B-Splines," Song Han and Gerard Medioni 943

"Dynamic Programming Delineation," Lee Iverson 951

"Finding the Perceptually Obvious Path," Martin A. Fischler 957

"Knowledge Directed Reconstruction from Multiple Aerial Images," Christopher O. Jaynes, Mauricio Marengoni, Allen Hanson, Edward Riseman, and Howard Schultz 971

"Recent Advances in 3D Reconstruction Techniques Using Aerial Images," Howard Schultz, Frank Stolle, Xiaoguang Wang, Edward M. Riseman, and Allen R. Hanson 977

"Site Modeling Using IFSAR and Electro-Optical Images," K.B. Hoepfner, Christopher Jaynes, Edward Riseman, Allen Hanson, and Howard Schultz 983

"Detection and Description of Buildings from Multiple Aerial Images," Sanjay Noronha and Ram Nevatia 989

"Ridge and Ravine Detection in Digital Images," Thomas C. Henderson, Scott Morris, and Charlotte Sanders 999

"Extraction of Micro-Terrain Ravines Using Image Understanding Constrained by Topographic Context," Gregory W. Thoenen and William B. Thompson 1001

"Quantitative Comparison of IU Algorithms," Eric S. Jensen and William B. Thompson 1007

SECTION V: AUTOMATIC TARGET RECOGNITION (ATR)

Automatic Target Recognition - Principal Investigator Reports

"Model-Based Target Recognition in Foliage Penetrating SAR Images," R. Chellappa , P. Burlina, and J. Song 1013

"Feature Extraction using Attributed Scattering Center Models for Model-Based Automatic Target Recognition," Randy Moses and Lee Potter 1019

"A Unified, Multiresolution Framework for Automatic Target Recognition," E. Grimson, J. Shapiro, P. Viola, and A. Willsky 1023

"Statistical Independent and Relevant Feature Extraction for Classification of SAR Imagery," Jose C. Principe 1029

"Context and Quasi-Invariants in ATR with SAR Imagery," Thomas O. Binford, By-Her Wang, and Tod S. Levitt 1031

"3D Object Recognition from Multiple and Single Views," Isaac Weiss and Azriel Rosenfeld 1041

"Wavelet-based Target Hashing for Automatic Target Recognition," Robert Hummel and Davi Geiger 1047

"Image Understanding Research for Battlefield Awareness at Johns Hopkins University," Lawrence B. Wolff 1057

"Image Understanding Research at UC Irvine: Automatic Recognition in Multispectral Imagery," Glenn Healey 1063

Automatic Target Recognition - Technical Papers

"Soft Competitive Principal Component Analysis Using The Mixture of Experts," Craig L. Fancourt and Jose C. Principe 1071

"A Nonparametric Methodology for Information Theoretic Feature Extraction," John W. Fisher III and Jose C. Principe 1077

"Toward a Fundamental Understanding of Multiresolution SAR Signatures," Gilbert Leung and Jeffrey H. Shapiro 1085

"Use of Context for False Alarm Reduction in SAR Automatic Target Recognition," S. Kuttikkad, W. Phillips, S. Mathieu-Marni, R. Meth, and R. Chellappa 1091

"A Parametric Model for Synthetic Aperture Radar Measurements," Mike Gerry, Lee Potter, and Randy Moses 1105

"Multiple Stochastic Models for Recognition of Occluded Targets in SAR Images," Bir Bhanu and Bing Tian 1119

"Scale-Based Robust Image Segmentation," A. Kim, I. Pollak, H. Krim, and A.S. Willsky 1129

"Invariants for the Recognition of Articulated and Occluded Objects in SAR Images," Grinnell Jones III and Bir Bhanu 1135

"Reinforcement Learning Integrated Image Segmentation and Object Recognition," Bir Bhanu, Xin Bao, and Jing Peng 1145

"Target Detection in UWB SAR Images Using Temporal Fusion," Li-Kang Yen and Jose C. Principe 1155

"A Self-organizing Principle for Segmenting and Super-resolving ISAR Images," Frank M. Candocia and Jose C. Principe 1161

"Matching of Articulated Objects in SAR Images," Joon Soo Ahn and Bir Bhanu 1167

"Lie Group Analysis in Object Recognition," D. Gregory Arnold, Kirk Sturtz, and Vince Velten 1173

"Monte Carlo Comparison of Distance Transform Based Matching Measures," Daniel P. Huttenlocher 1179

"Photometric Computation of the Sign of Gaussian Curvature Using a Curve-Orientation Invariant," Elli Angelopoulou and Lawrence B. Wolff 1185

"Face k-D Trees for Bounded Error Point Location Operations and Surface Simplification," James P. Williams and Lawrence B. Wolff 1191

"Experiments on (Intelligent) Brute Force Methods for Appearance-Based Object Recognition," Randal C. Nelson and Andrea Selinger 1197

"Recognizing Objects by Matching Oriented Points," Andrew Edie Johnson and Martial Hebert 1207

"Medialness and Skeletonization for Object Registration and Shape Similarity," Andrew Bzostek and Lawrence B. Wolff 1219

"Hierarchical Silhouettes Classification using Curve Matching," Yoram Gdalyahu and Daphna Weinshall 1223

"Appearance Based Object Recognition with Illumination Invariance," Kohtaro Ohba, Yoichi Sato, and Katsushi Ikeuchi 1229

"Generic, Model-Based Edge Estimation in the Image Surface," Pei-Chun Chiang and Thomas O. Binford 1237

"Automated Construction of Templates for Matching," Gang Liu and Robert M. Haralick 1247

"Performance Modeling and Adaptive Target Detection," David M. Doria 1255

"Linear Models for Infrared Spectra," Glenn Healey and Luis Benites 1267

"Recognition Using Multiband Filtered Energy Matrices," Lizhi Wang and Glenn Healey 1273

"Using Spectral/Spatial Information for Automatic Recognition," Glenn Healey and David Slater 1281

"A Neural Network Approach to Indexing," Mark R. Stevens, J. Ross Beveridge, and Charles W. Anderson 1289

"Evaluations of Large, Complex Research and Development Programs: Theory and Practice," Theodore R. Yachik and Lynne Gilfillan 1291

SECTION VI: MULTIDISCIPLINARY UNIVERSITY RESEARCH INITIATIVE (MURI)

Multidisciplinary University Research Initiative - Principal Investigator Reports

"A Trainable Modular Vision System," R. Brooks, E. Grimson, T. Poggio, C. Koch, C. Sodini, L. Stein, and W. Yang 1307

"Multidisciplinary Image Understanding Research at the University of Maryland," Azriel Rosenfeld 1315

"Advanced Visual Sensor Systems," Terrance E. Boult, Rick Blum, Richard Wallace, Gary Zhang, Shree K. Nayar, Peter K. Allen and John R. Kender 1323

"Integrated Vision and Sensing for Human Sensory Augmentation," Takeo Kanade and Vladimir Brajovic 1335

"Principal Investigator Report: Automated Vision and Sensing Systems at Boston University," Stephen Grossberg, Gail Carpenter, Eric Schwartz, Ennio Mingolla, Daniel Bullock, Paolo Gaudiano, Andreas Andreou, Gert Cauwenberghs, and Allyn Hubbard 1345

Multidisciplinary University Research Initiative - Technical Papers

"Model-Based Matching by Linear Combinations of Prototypes," Michael J. Jones and Tomaso Poggio ... 1357

"Orientation Behavior Using Registered Topographic Maps," C. Ferrell 1367

"A Bootstrapping Algorithm for Learning Linear Models of Object Classes," Thomas Vetter Michael J. Jones, and Tomaso Poggio 1373

"Desktop Programmable Pixel-Parallel Accelerator for High Speed Image Processing," J.C. Gealow, N.S. Love, G. Hall, I Masaki, and C.G. Sodini 1379

"Hardware for Content-Based Image Queries," Aaron Lipman and Woodward Yang 1385

"A Coarse-grained, Reconfigurable Image Coprocessor," Alexander Bugeja and Woodward Yang 1389

"A Perspective 3D Formalism for Shape from Shading," Isaac Weiss 1393

"Image Segmentation and Labeling Using the Polya Urn Model," A. Banerjee, P. Burlina, and F. Alajaji ... 1403

"Analysis of Reconstruction from Multiple Views," Cornelia Fermuller and Yiannis Aloimonos 1411

"Reflectance and Texture of Real-World Surfaces," Kristin J. Dana, Shree K. Nayar, Bram van Ginneken, and Jan J. Koenderink 1419

"Parametric Feature Detection," Simon Baker, Shree K. Nayar, and Hiroshi Murase 1425

"Catadioptric Image Formation," Shree K. Nayar and Simon Baker 1431

"Imaging-Consistent Super-Resolution," Ming-Chao Chiang and Terrance E. Boult 1439

"Multisensor Image Fusion Using a Region-Based Wavelet Transform Approach," Zhong Zhang and Rick S. Blum 1447

"A Variable Neighborhood Approach to Early Vision," Yuri Boykov, Olga Veksler, and Ramin Zabih 1453

"Toward Automatic Domain-Adaptation in Artificial Evolution: Experiments with Face Recognition," Matthew R. Glickman and Katia P. Sycara 1459

"Visual Learning for Landmark Recognition," Yutaka Takeuchi, Patric Gros, Martial Hebert, and Katsushi Ikeuchi 1467

"Evolutionary Learning for Orchestration of a Signal-to-Symbol Mapper," Astro Teller and Manuela Veloso 1475

"Multi-Spectral Imaging Filters," L.J. Denes, M. Gottlieb, B. Kaminsky, P. Metes, Z.K. Kun, M. Capizzi, J. Hibner, D. Purta, and A.M. Guzman 1483

"Automated Vision and Sensing Systems at Boston University," Stephen Grossberg, Gail Carpenter, Eric Schwartz, Ennio Mingolla, Daniel Bullock, Paolo Gaudiano, Andreas Andreou, Gert Cauwenberghs, and Allyn Hubbard 1491

AUTHOR INDEX

Ahuja, Narendra 129, 465, 537, 541
Alajaji, F. 1403
Allen, Peter K. 921, 1323
Aloimonos, Yiannis 1411
Amabile, M.J. 185
Anandan, P. 3, 11, 639
Anderson, Charles W. 1289
Andreou, Andreas 1345, 1491
Angelopoulou, Elli 1185
Araujo, Helder 875
Arnold, D. Gregory 1173
Asmuth, Jane C. 849
Avidan, Shai 863
Azarbayjani, Ali 193

Bajcsy, Peter 541
Baker, Simon 1425, 1431
Balakirsky, Stephen 285
Balasubramaniam, Karthik 275
Banerjee, A. 1403
Bao, Xin 1145
Barrett, Eamon B. 407
Barrett, Tom 455
Benites, Luis 1267
Beveridge, J. Ross 275, 825, 1289
Bhanu, Bir 483, 1119, 1135, 1145, 1167
Binford, Thomas O. 513, 835, 1031, 1237
Blum, Rick S. 1323, 1447
Bobick, Aaron 25, 227
Bolles, Robert C. 41, 325, 759
Boult, Terrance E. 55, 1323, 1439
Bowling, Michael 779
Boyd, Jeffrey 303
Boykov, Yuri 1453
Brajovic, Vladimir 177, 1335
Brand, Matt 193
Bremner, Bill 381
Brill, Frank Z. 159
Brooks, R. 1307
Brown, Christopher 69, 875
Bugeja, Alexander 1389
Bullock, Daniel 1345, 1491
Bulwinkle, G. Edward 779
Burlina, Phillipe 415, 529, 577, 1013, 1403
Burt, Peter 3, 11
Bzostek, Andrew 1219

Candocia, Frank M. 1161
Capizzi, M. 1483
Carceroni, Rodrigo L. 875
Carpenter, Gail 1345, 1491
Cauwenberghs, Gert 1345, 1491
Chellappa, Rama 19, 247, 285, 295, 415, 577 615, 1013, 1091
Chiang, Ming-Chao 1439
Chiang, Pei-Chun 1237
Chou, George T. 869
Cochran, Steven Douglas 779
Cohen, Scott D. 669
Collins, Robert T. 3
Connolly, Christopher I. 585
Coorg, Satyan 857
Curwen, R.W. 595

Dana, Kristin J. 1419
Das, Madirakshi 701
Davis, Larry 19, 113, 135, 415
De Bonet, Jeremy S. 655
de St. Germain, H. James 927
Denes, L.J. 1483
Dolan, John 723
Doria, David M. 475, 1255
Draper, Bruce A. 825
Duric, Zoran 143
Dyer, Charles R 63, 881, 935

Fancourt, Craig L. 1071
Faugeras, O. 45
Fejes, Sandor 113
Fermuller, Cornelia 1411
Ferrell, C. 1367
Finci, Ilan 255
Fischler, Martin A. 41, 325, 759, 957
Fisher, John W. III 1077
Flinchbaugh, Bruce 51
Ford, Stephen J. 779

Gaudiano, Paolo 1345, 1491
Gdalyahu, Yoram 1223
Gealow, J.C. 1379
Geiger, Davi 1047
Gerry, Mike 1105
Gerson, Donald J. 401
Gilfillan, Lynne 1291
Glatz, William R. 401
Glickman, Matthew R. 1459
Gottlieb, M. 1483
Govindu, Venu 615
Graves, Christopher 275
Grimson, E. 45, 625, 675, 681, 1023, 1307
Gros, Patrick 1467
Grossberg, Stephen 1345, 1491
Guibas, Leonidas J. 495, 661, 669
Guzman, A.M. 1483

Hackett, Doug 381, 455
Hall, G. 1379
Hamilton, D. 503
Hampshire, John B. II 449, 547
Han, Song 943

Hannah, Marsha Jo 325
Hanson, Allen R. 813, 971, 977, 983
Haralick, Robert M. 1247
Hartley, Richard I. 631, 649
Harvey, Wilson A. 779
Healey, Glenn 1063, 1267, 1273, 1281
Hebert, Martial 1207, 1467
Heller, Aaron J. 585, 759
Henderson, Thomas C. 819, 927, 999
Herman, Joshua 261
Hibner, J. 1483
Hoepfner, K.B. 983
Hoogs, Anthony 381, 455, 565, 607
Huang, Jing 687
Huang, Thomas 465
Hubbard, Allyn 1007, 1345, 1491
Huertas, A. 437
Hummel, Robert 1047
Hunter, Edward 303
Hussain, Mushtaq 325
Huttenlocher, Daniel P. 89, 1179

Ikeuchi, Katsushi 95, 911, 1229, 1467
Irani, Michal 79, 639
Iverson, Lee 951

Jain, Ramesh 85, 303
Jaynes, Christopher O. 983, 971
Jensen, Eric S. 1007
Jochem, Todd 345
Johnson, Andrew Edie 1207
Jones, Grinnell III 1135
Jones, Michael J. 1357, 1373

Kalp, Dirk 779
Kaminsky, B. 1483
Kanade, Takeo 3, 95, 177, 357, 901, 1335
Kelly, Patrick 303
Kender, John R. 221, 1323
Kim, A. 1129
Kjeldsen, Rick 221
Koch, C. 1307
Koenderink, Jan J. 1419
Krim, H. 1129
Kumar, Rakesh 265, 849
Kun, Z.K. 1483
Kurt G. Konolige 41
Kuttikkad, S. 1091

Langley, Pat 835
Lanzillotto, A.M. 185
Leclerc, Yvan G. 431, 585
Lerner, Richard A. 715, 723
Lesher, Christopher E. 275
Leu, T.S. 185
Leung, Gilbert 1085
Leventon, M.E. 625
Levitt, Tod S. 1031
Lipman, Aaron 1385
Lipson, Pamela R. 675
Lipton, Alan J. 3
Liu, Andrew 201
Liu, Gang 1247
Love, N.S. 1379
Lozano-Perez, T. 45, 893
Lundberg, Andrew 375
Luong, Q.-T. 519
Luong, Tuan 325

Maloof, Marcus A. 835
Manmatha, R. 693, 707
Marengoni, Mauricio 971
Marjanovic, M. 215
Masaki, I. 1379
Mathieu-Marni, S. 1091
McGlone Chris 779
McKeown, David M. Jr. 779
McMahill Jeff 779
Medioni, Gerard 31, 943
Mellor, J.P. 893
Metes, P. 1483
Meth, R. 1091
Mingolla, Ennio 1345, 1491
Moisan, Sabine 529
Morimoto, Carlos 285, 295
Morris, Scott 707, 999
Moses, Randy 1019, 1105
Mundy, Joseph L. 407, 425, 503, 595
Munjy, Riadh 325
Murase, Hiroshi 1425

Nayar, Shree K. 55, 235, 243, 1323, 1419, 1425, 1431
Nelson, Randal 69, 1197
Nevatia, Ram 31, 437, 771, 989
Noronha, Sanjay 989

Ohba, Kohtaro 1229
Oliver, Nuria 193
Olson, Thomas J. 159
Oren, Michael 207
Osuna, Edgar 207

Papageorgiou, Constantine 207
Parameswaran, V. 577
Payton, Paul M. 407
Peleg, Shmuel 79, 255, 261
Peng, Jing 1145
Pentland, Alex 25, 193, 201
Peri, Venkata N. 243
Phillips, W. 1091
Pinhanez, Claudio 227
Poggio, T. 45, 25, 207, 1357, 1373, 1307
Polis, Michael F. 779
Pollak, I. 1129
Pomerleau, Dean 339, 345
Potter, Lee 1019, 1105
Price, Keith 771
Principe, Jose C. 1029, 1071, 1077, 1155, 1161
Purta, D. 1483
Puscar, Michael A.J. 607

Rao, Rajesh P.N. 123
Ratakonda, Krishna 537
Ratan, A. Lakshmi 681
Ravela, S. 693
Reed, Michael K. 921
Riseman, Edward 971, 701, 707, 813, 977, 983
Rivlin, Ehud 143
Rosenberg, Yoav 153
Rosenfeld, Azriel 143, 1041, 1315
Rousso, Benny 255
Rubner, Yossi 661

Sage, Stephanie 835
Sanders, Charlotte 707, 999
Sato, Yoichi 911, 1229
Sawhney, Harpreet S. 265, 849
Saxena, Tushar 649
Scassellati, B. 215
Schlenzig, Jennifer 303
Schultz, Howard 813, 971, 977, 983
Schwartz, Eric 1345, 1491
Seitz, Steven M. 881, 935
Selinger, Andrea 1197
Shapiro, Jeffrey H. 1023, 1085
Shashua, Amnon 79, 863, 889
Shekhar, Chandra 415, 529, 615
Shufelt, Jefferey A. 779
Simon, David A. 901
Sinha, Pawan 207, 675
Slater, David 1281
Smith, Michael A. 357
Snell, V. 503
Sodini, C.G. 1307, 1379
Song, J. 1013
Soo Ahn, Joon 1167
Srinivasa, Narayan 129
Srinivasan, S. 247
Stark, Stevan R. 927
Stein, G.P. 889
Stein, L. 1307
Stevens, Mark R. 1289
Stewart, C. 503
Stolle, Frank 977
Sturtz, Kirk 1173
Sycara, Katia P. 1459

Tai, Andy 303
Takeuchi, Yutaka 1467
Teller, Astro 1475
Teller, S. 45, 767, 857, 869, 893
Thoenen, Gregory W. 1001
Thompson, William B. 819, 927, 1007, 1001
Thorpe, Chuck 367
Tian, Bing 1119
Tomasi, Carlo 495, 661

van Ginneken, Bram 1419
Veksler, Olga 1453
Veloso, Manuela 1475
Velten, Vince 1173
Vetter, Thomas 1373
Viola, Paul 45, 655, 1023

Wallace, Richard 1323
Wang, By-Her 513, 1031
Wang, Lizhi 1273
Wang, Xiaoguang 977
Weinshall, Daphna 79, 1223
Weiss, Isaac 1041
Weiss, Isaac 1393
Wells III, W.M. 625
Werman, Michael 79, 153
Wheeler, Mark D. 911
Wildes, R.P. 185
Williams, James P. 1191
Williamson, M. 215
Willsky, A.S. 1023, 1129
Wixson, Lambert 3
Wolff, Lawrence B. 375, 1057, 1185, 1191, 1219
Wood, Sidney E. 401
Wu, Victor 707

Yachik, Theodore R. 1291
Yacoob, Yaser 19, 135
Yang, W. 1307
Yang, Woodward 1385, 1389
Yen, Li-Kang 1155
Yocum, Daniel 779

Zabih, Ramin 89, 687, 1453
Zhang, Gary 1323
Zhang, Zhong 1447
Zheng, Qinfen 19, 415

FOREWORD

Those who are obsessed with practice, but have no science,
are like a pilot out with no tiller or compass.
—Leonardo da Vinci

This Proceedings

This proceedings contains the assembled reports of the various research projects that comprise the DARPA Image Understanding (IU) Program. Submissions from forty academic and industrial computer vision research laboratories document progress and lessons learned in research performed for applications in image registration, target recognition, image exploitation, cartography, 3D model reconstruction, video surveillance, activity recognition and content-based image retrieval.

This is the 25th proceedings in the series, which has become known as a comprehensive source for the latest research results in image understanding from the nation's leading IU laboratories. Like its predecessors, this proceedings is not peer-reviewed in the traditional sense of a refereed conference or journal. Instead, the principal investigator of each laboratory is responsible for selecting the papers that will represent the work carried out in his lab. Because the reputation of each lab is at stake, the quality of submissions has remained consistently high.

The Image Understanding Workshop

The 1997 Image Understanding Workshop is being held at a pivotal moment in the history of the IU Program. The major application projects of the last five years, the RADIUS program for research in automated image exploitation, and the Reconnaissance, Surveillance, and Target Acquisition (RSTA) Program in support of the DARPA Unmanned Ground Vehicle Program, have been completed, and the new battlefield awareness application thrusts for the IU Program have just been launched. Accordingly, the IU Workshop, and this proceedings, reports both on previous accomplishments as well as plans for the new projects.

The RADIUS Program has pioneered a new paradigm for performing automated image examination – called *model supported exploitation* (MSE). Rather than attempt to detect changes based on pixel differencing or comparison of extracted features, the model-supported exploitation approach calls for the construction and use of detailed context models that sufficiently constrain the computer vision tasks so as to make them tractable. This approach has been used effectively within the RADIUS Testbed System, which has been installed at the National Photographic Interpretation Center and is undergoing a series of evaluations by image analysts. RADIUS has spawned a number of related activities (SIAS, FOCUS, Pinpoint, CrossCut, SMS, BCAMS, APGD) that are applying the MSE paradigm to a variety of image interpretation tasks. A comprehensive description of the many facets of RADIUS has been published as a book, *RADIUS: Image Understanding for Imagery Intelligence*, available from the Morgan Kaufmann publishing company.

The RSTA Program has developed several new technologies for the reconnaissance payload of the DARPA Unmanned Ground Vehicle. Image stabilization, sensor planning, exploitation of polarized light, and target recognition for infrared, LADAR, and color imagery were exhibited in demonstrations and field exercises conducted as part of the Demo II UGV Program. Formal evaluation of target detection and recognition algorithms has been performed at the US Army Night Vision Electronic Systems Directorate at Fort Belvoir. Results of these evaluations as well as detailed descriptions of the algorithms and lessons learned are now available as the book, *Reconnaissance, Surveillance, and Target Acquisition for the Unmanned Ground Vehicle: Providing Surveillance Eyes for an Autonomous Vehicle*, also published by Morgan Kaufmann.

The New Image Understanding Program

The mission of the IU Program is to advance the state-of-the-art in computer vision so as to enable new applications of imaging technology in support of the warfighters. Toward this end, the IU Program sponsors fundamental research, applications development, and supporting infrastructure.

Research and Applications

Research to strengthen the theoretical foundations of image understanding accelerates the development of subsequent IU applications. Because there are an infinite variety of promising avenues to explore, the IU Program has chosen to investigate those that are both highly promising *and* relevant to battlefield awareness. In particular, IU research is focused on the following applications

IMEX: Image Exploitation, to include automatic target recognition, develops novel ways to use high revisit multiple sensor imagery from unmanned aerial vehicles and other reconnaissance platforms for change detection, site monitoring, and activity tracking. The Site Monitoring System, a component of the Semiautomated Imagery Processing (SAIP) system, is a major focus for this research.

APGD: Automatic Population of Geospatial Databases seeks to increase the level of automation used for constructing geospecific 3D models for simulation, mission rehearsal, and intelligence applications.

VSAM: Video Surveillance and Monitoring develops video understanding technology to be used for urban and battlefield surveillance where human observation is too costly, dangerous, or otherwise impractical.

Infrastructure

RCDE: The RADIUS Common Development Environment is in use at many IU laboratories, and is being used within the RADIUS, SMS, and APGD programs. It has recently been ported to Silicon Graphics computers, and provides a high-performance environment for development of 3D image exploitation applications.

IUE: The Image Understanding Environment continues to advance rapidly, both in scope and maturity of its implementation. The IUE, along with its libraries of algorithms contributed by the IU community, will become a powerful tool enabling the exchange of algorithms and results.

IU Data Server: An image server has been established at the Air Force Wright Laboratories to facilitate the provision of imagery for use on IU contracts. Greatly expanded volumes of imagery, now available unclassified and without export restrictions, will stimulate research on techniques that are relevant to real-world situations.

Current descriptions of all activities within the DARPA Image Understanding Program are maintained on the World Wide Web: http://www.hokie.bs1.prc.com/iu/iuhome.htm

The Role of Research

It has become popular within the government nowadays to identify technologies that are available for immediate application and to transition them to end users as rapidly as possible. This practice, commonly referred to as "panning for gold" or "picking the low-hanging fruit," can maximize the payoff of previous investments in research and technology development. While valuable as a means to transition current technology, this activity must not come at the expense of continued investment in research and development, as doing so would ultimately jeopardize the nation's ability to field new technology in the years ahead.

The DARPA Image Understanding Program has traditionally stimulated the computer vision research community – the existence of the fielded applications described at this workshop (as well as many others) attest to the wisdom of those earlier investments. The IU Program continues to maintain its focus on those high-payoff, high-risk research topics that hold promise for enabling future systems development programs. This continued investment in research is putting the fruit on the trees for others to pick in the years ahead.

Thomas M. Strat
Image Understanding Program Manager
Defense Advanced Research Projects Agency

ACKNOWLEDGEMENTS

The twenty-fifth Image Understanding (IU) Workshop was held in New Orleans, Louisiana on May 12-14, 1996. The workshop was sponsored by the Information Systems Office (ISO) of the Defense Advanced Research Projects Agency (DARPA). Co-chairs of this years conference were Dr. Bruce Flinchbaugh, Professor Rama Chellappa, and Professor Aaron Bobick. They were responsible for the organization of the technical program.

Two special sessions were held this year. The Model-Supported Exploitation Session presented six system development programs that are spinoffs of the model-supported exploitation technology pioneered in the RADIUS Program. The Small Business Innovative Research Session presented six ongoing activities aimed at commercializing IU technology.

Executive agents from many government organizations are actively involved with the IU program. Officials of NIMA, TEC, ARL, WL, NVESD, ONR, ORD, AFOSR, ARO, and RL are active participants in the program and have contributed significantly to progress in image understanding, particularly in transferring the research results to development programs.

The cover of the proceedings was designed by Bruce Flinchbaugh of Texas Instruments and illustrates the three application foci of the restructured IU Program. Dave Stenger of PRC produced the graphics used on the cover and for all meeting materials. Morgan Kaufmann Publishers, Inc. has performed its customary excellent services as publisher of this proceedings.

The workshop coordinators were Andrea Jones and Sheryl Augustson of PRC, Inc. assisted by Lisa Austin and Kelli Sakata. Together, they were responsible for planning and coordinating the thousands of large and small details that go into putting this event together. Their diligence and teamwork resulted in the excellent arrangements enjoyed by all who attended.

Steve Hennessy of SAIC contributed to the program formulation. Lois Hollan of PRC assisted and supervised all aspects of the 1997 Image Understanding Workshop.

SECTION I
VIDEO SURVEILLANCE AND MONITORING (VSAM)

VIDEO SURVEILLANCE AND MONITORING (VSAM) PRINCIPAL INVESTIGATOR REPORTS

Cooperative Multi-Sensor Video Surveillance*

Takeo Kanade, Robert T. Collins, and Alan J. Lipton
Carnegie Mellon University, Pittsburgh, PA
E-MAIL: {kanade,rcollins,ajl}@cs.cmu.edu
HOMEPAGE: http://www.cs.cmu.edu/~vsam

P. Anandan, Peter Burt, and Lambert Wixson
David Sarnoff Research Center, Princeton, NJ
E-MAIL: {panandan,pburt,lwixson}@sarnoff.com

Abstract

Carnegie Mellon University (CMU) and the David Sarnoff Research Center (Sarnoff) have begun a joint, integrated feasibility demonstration in the area of Video Surveillance and Monitoring (VSAM). The objective is to develop a cooperative, multi-sensor video surveillance system that provides continuous coverage over large battlefield areas. Image Understanding (IU) technologies will be developed to: 1) coordinate multiple sensors to seamlessly track moving targets over an extended area, 2) actively control sensor and platform parameters to track multiple moving targets, 3) integrate multi-sensor output with collateral data to maintain an evolving, scene-level representation of all targets and platforms, and 4) monitor the scene for unusual "trigger" events and activities. These technologies will be integrated into an experimental testbed to support evaluation, data collection, and demonstration of other VSAM technologies developed within the DARPA IU community.

1 Introduction

The recent growth in diverse imaging sensors and deployment platforms opens exciting new possibilities for Video Surveillance and Monitoring (VSAM) systems that provide continuous battlefield awareness. Future military scenarios will involve multiple sensors mounted on maneuverable ground and air vehicles cooperating with stationary ground sensors to monitor large battlefield areas for enemy troop movements (Figure 1).

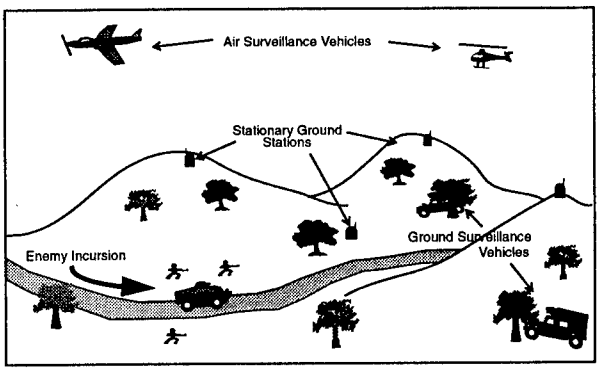


Figure 1: Multiple sensors cooperate to provide broad battlefield coverage.

Carnegie Mellon University (CMU) and the David Sarnoff Research Center (Sarnoff) have begun an integrated feasibility demonstration (IFD) to develop image understanding (IU) technologies to support this cooperative, multi-sensor, battlefield VSAM scenario. This report describes the overall objectives of the CMU-Sarnoff VSAM IFD project, their relevance to battlefield situational awareness, the key scientific and technology challenges to be addressed, and plans for the development, demonstration, and evaluation of the new VSAM technologies.

2 Objectives and Military Relevance

The major object of the CMU-Sarnoff IFD team is to develop a suite of VSAM technologies that enable a single human operator at a worksta-

*This work is funded under DARPA BAA 96-14.

tion to supervise a network of remote VSAM platforms (stationary, moving on the ground, or airborne), having multiple, steerable sensors. Platform surveillance operations will be mainly autonomous, notifying the operator only of salient information as it occurs, and engaging the operator minimally to alter platform operations. The network of sensors will cooperate to perform broad-area monitoring and continuous target tracking over large areas that can not be viewed continuously by a single sensor alone. The IFD team will integrate this technology suite into an experimental testbed system that will additionally support evaluation, data collection, and demonstration of other VSAM technologies developed within the DARPA IU community.

Cooperative multi-sensor surveillance will significantly enhance battlefield awareness, by providing the commander with complete and continuous coverage of troop movements and target activities within a broad area. Examples of military scenarios that can use the VSAM technologies include:

- *perimeter monitoring*, in which a continuous watch is maintained over a familiar facility such as a warehouse, a military base, or a sensitive building. The major objectives of the monitoring task are to be alert to potential incursions by enemy troops or other suspicious activity,
- *forward observer*, in which ground and air-based surveillance vehicles are sent ahead of the troops to determine potential hazards for intended troop movements,
- *border patrol*, in which border areas are monitored for potential drug and/or weapon trafficking,
- *point reconnaissance* of a location such as a bridge, weapon storage site, an entry gate, or a suspected terrorist hangout for unusual movements and loitering by people or vehicles, and
- *cantonnement facility monitoring*, in which video observations of a weapons cantonnement facility collected over multiple days are analyzed to detect potential weapon movements.

The prototype testbed system that will be developed by the CMU-Sarnoff team will facilitate growth in the area of VSAM IU by supporting development and evaluation of component technologies. Potential military users will be able to observe field demonstrations, guide the selection of problems, and provide feedback on the utility of the developed components. In the optional out-years of the program, an integrated system will be delivered for testing and evaluation by military users, enabling the transfer of VSAM technologies to the DOD community.

In addition to the military applications mentioned above, this effort will also spur technology transfer to commercial applications, such as building and parking lot security, warehouse guard duty, and monitoring restricted access areas in airports. Combined ground and air surveillance capabilities also have promising applications in civilian law-enforcement operations.

3 Scientific and Technical Challenges

The major scientific and technical challenges of the CMU-Sarnoff VSAM approach are to: 1) coordinate multiple sensors to seamlessly track moving targets over an extended area in a visually complex environment, 2) actively control sensor and platform parameters to track multiple moving targets, 3) provide scene-level representations of targets and their environment by integrating evolving visual, geometric, and symbolic sensor observations together with collateral scene data, and 4) monitor the scene for unusual "trigger" events and activities that should cue further processing or operator involvement. This section outlines the technical challenges that IFD research must address in order to meet the above objectives.

3.1 Coordinating multiple sensors

Central to the goal of the VSAM IFD program is real-time detection and tracking of targets over a wide area using multiple distributed sensors. To perform this task, the following technical areas will be addressed. Note that all of the operations described must be performed in real-time.

Robust target detection and tracking: Targets must be detected and continuously followed as they move through a large cluttered area, even when they disappear behind occluding surfaces and later reappear, or when they stop and later resume moving. Tracking must be maintained as the camera pans, tilts, and zooms in to obtain a closer look, and in the presence of image motion containing 3D parallax induced by movement of the sensor platform. A combination of motion and appearance cues will be used to achieve robust target tracking.

Continuous target following using multiple distributed sensors: Targets must be continuously followed as they move out of the field-of-view of one sensor into that of another. This requires establishing the correspondence of the fields-of-views of the different cameras to achieve target "hand-off". It also requires appearance matching of the target as seen by sensors with significantly different viewpoints.

Cooperative ground-and-air surveillance: Targets detected in airborne views can be used to cue local ground sensors, and vice versa. This requires geo-registering airborne views with a set of ground-based views. In order to achieve the geolocation accuracy required for air-to-ground (or ground-to-air) hand-off, visual pose refinement using cultural landmarks and terrain features will be performed to refine initial pose estimates based on platform ephemeris data.

3.2 Active sensor control

Active camera control will be performed to maximize system performance and maintain target pursuit over large areas. This involves controlling sensing parameters (e.g. view direction, zoom, panning speed, vergence angles), processing resources (resolution, focus of attention, load balance), and mobile sensor deployment.

Sensor planning and control: Sensor hand-off for cooperative, multi-sensor surveillance will be achieved using standard visibility and occlusion analysis. This requires using collateral terrain maps and 3D site models containing man-made features to perform visibility analysis from each sensor position, to determine, based on current estimates of target trajectory, which sensor will have the closest, unoccluded view. This work will also involve task-based planning of new camera views, while imposing physical constraints on sensor platform mobility.

Multi-tasking for multiple target tracking: Occassionally, a single camera resource must be used to track multiple moving objects, not all of which fit within a single field of view. This problem will be addressed by introducing sensor multi-tasking, meaning that the camera field of view will be periodically switched between two (or more) targets that are being monitored. This requires continuously locating and updating the target positions within a panoramic reference mosaic image or a map, and using a combination of visual and inertial information to perform the scans.

3.3 Scene-level representation

An important component of the VSAM testbed is an interface that allows the human operator to visualize all available scene information, and to control the sensor suite to achieve mission objectives. To do this, information from multiple sensors will be integrated with collateral site information to provide an evolving scene-level representation (Figure 2).

Multi-sensor information integration: Information in the form of estimated target locations and appearances will be gathered from many difference sensors, possibly of different sensing modalities, and redundant data must be correlated and merged. This will be handled by transforming all target and platform locations and trajectories into a georeferenced coordinate system, either by locating them with respect to calibrated reference imagery, or solving for 3D position directly using known constraints such as terrain elevation.

Dynamic scene visualization: Comprehending a vast flow of incoming information from multiple sensors, regarding multiple targets, is a challenging task for any human operator. To make the task easier, a comprehensive, graphical visualization of the dynamic scene will be presented to the user that combines elements of

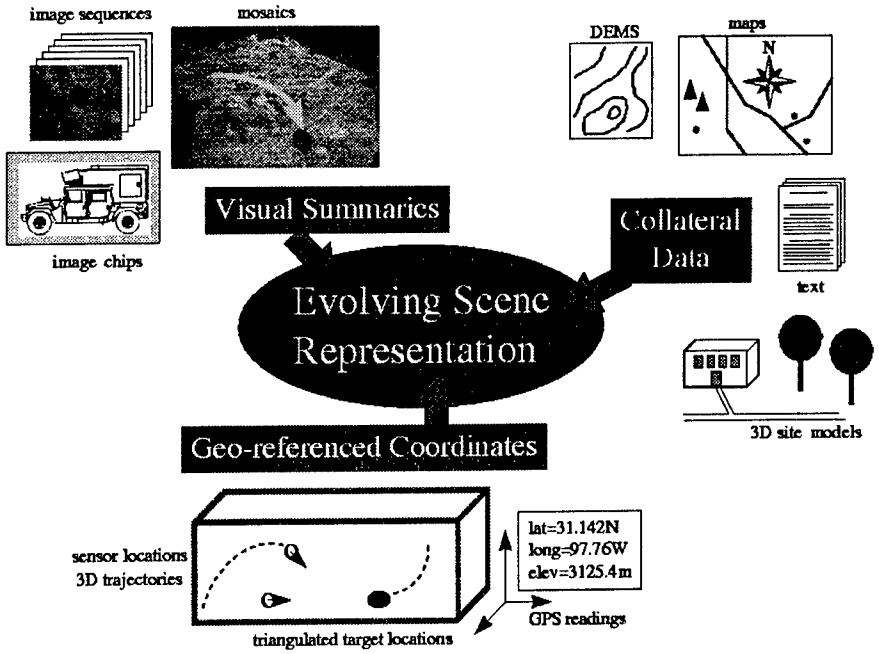


Figure 2: Components of an evolving, dynamic, scene-level representation.

visual sensor imagery, prior geometric models of the scene and targets, other collateral information such as maps, and symbolic depictions of activities of interest.

Collateral data integration and update: Prior collateral information about the scene will be maintained in the form of annotated maps, digital elevation models, reference imagery (e.g. satellite photos) and symbolic 3D site models. These will all be tied to the common geospatial scene coordinate frame. Incoming imagery will be used to not only update the positions of dynamic targets in the evolving scene model, but also to refine these prior models based on close-range views from the deployed sensor platforms.

3.4 Activity Monitoring

By broadening the scope of VSAM technology beyond simple 2D image-level tracking into dynamic, scene-level descriptions co-registered with 3D collateral data, the CMU-Sarnoff approach will enable research into high-level activity and event monitoring. For example, the system could be tasked to monitor sensitive areas for such "suspicious" activities as:

- vehicles going the wrong way down a one-way street,
- vehicles (or people) entering a restricted access area,
- vehicles that repeatedly circle the block around a sensitive building,
- people coming and going from the front door of a suspected drug hideout,
- pedestrians who loiter in front of a building for a long time,
- pedestrians trying to look over a fence, or peer through windows.

Many of these tasks would be difficult, if not impossible, to perform with 2D visual image data alone, but are enabled by having co-registered scene models to provide regions of interest and expected patterns of motion.

4 The VSAM testbed

The CMU-Sarnoff team is developing a testbed architecture that will support the design, evaluation, and demonstration of VSAM IU technologies developed by the IFD team and the rest of the DARPA VSAM community. The testbed architecture consists of multiple sensor processing units (SPUs) in the field, communicating with an operator control unit (OCU) connected to an operator console (see Figure 3). The goal

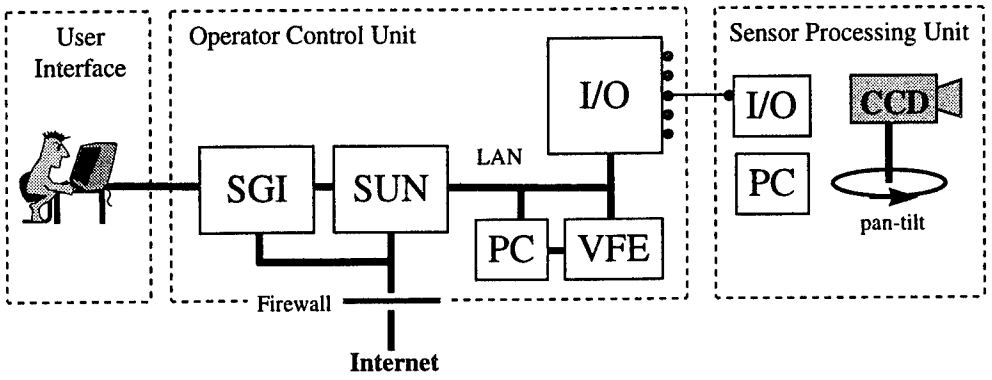


Figure 3: The VSAM testbed architecture

has been to design a testbed that is both rich enough (in terms of equipment and computational power) and flexible enough (in terms of functionality) to support a wide range of VSAM research.

Sensor processing units.

Multiple sensor processing units (SPUs) are mounted in fixed locations on hills or rooftops to provide distributed coverage over a wide area. At least two sensors will be mounted on mobile platforms – one ground vehicle (NavLab), and one airborne vehicle (autonomous helicopter or chartered flight).

The specification of what constitutes an SPU is intentionally left open-ended within the testbed architecture, in order to encompass a wide variety of sensor types such as monocular visible light and IR cameras, stereo heads, LADAR, and acoustic sensors. However, a typical SPU will consist of a color CCD camera with a motorized zoom lens, mounted on a controllable pan-tilt head. An onboard controller (e.g. Pentium PC) is responsible for collecting and managing sensor data, communicating with the OCU and generating the appropriate signals to control sensor hardware. Sensors mounted on mobile platforms will have access to real-time video processing hardware (Sensar VFE) for frame-rate video stabilization, and to onboard pose sensors for providing estimates of SPU location and orientation.

Operator Control Unit.

The operator control unit (OCU) is responsible for integrating the results produced from multiple sensors with a database of collateral scene information, in order to form and maintain an evolving, dynamic scene representation. The core of the OCU consists of two workstations (SGI and/or Sun), one dedicated primarily to the graphical user interface and the other handling information fusion and tasking control. Input from sensors in the field comes in via communication links ranging from radio ethernet and cell phone for discrete packets of symbolic information, to microwave links and coax cable for higher-bandwidth transmission of video streams. A Sensar VFE real-time video processor controlled by a PC host is provided to stabilize video streams from sensors that don't have enough onboard processing power. A local area network (LAN) connects all components to each other, and to an external internet connection, protected by a firewall.

Graphical User Interface.

One of the technical goals of the VSAM project is to demonstrate that a single human operator can effectively monitor a large battlefield area. Towards this end, the test system will have a graphical user interface for battlefied visualization and sensor suite tasking. Through the inerface, the operator can task individual sensor units, as well as the entire testbed sensor suite, to perform surveillance operations such as generating a quick summary of all target activities in the area. The operator may choose to see a map of the area, with all target and sensor platform locations overlaid on it. Alternatively, the operator may select a more immersive display

(with a more limited field of view) by interacting with a texture-mapped 3D model of terrain and cultural features (buildings and roads), within which dynamically updated sensor and target locations are displayed (Figure 4).

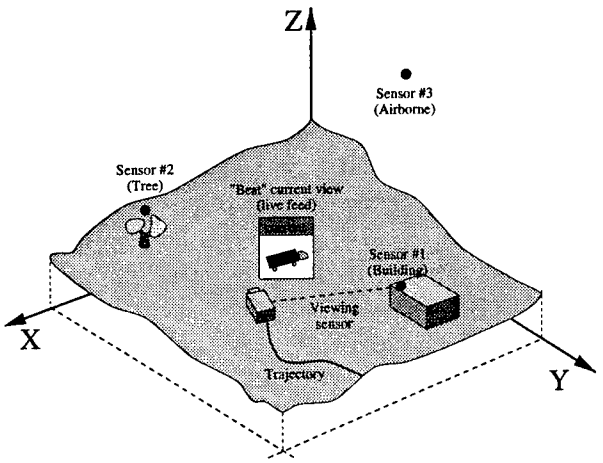


Figure 4: Sample user interface visualization.

The user interface will minimize operator typing by employing a graphical screen interface with "hot" areas that can be selected with a mouse or touch screen. For example, pointing to a sensor icon on the screen could bring up an overlay window showing the stabilized video output from that sensor viewpoint.

5 Demonstration Plan

Technology developed under the VSAM program will be demonstrated to the user community and the DARPA IU community through annual demonstrations. The Year 1 demonstrations will emphasize individual ground and air-based surveillance capabilities, whereas the Year 2 demonstration will emphasize combined ground and air surveillance. The Year 1 demonstrations are described in more detail below.

5.1 The Bushy-Run Site

The CMU "Bushy Run" site is a decommissioned chemical and nuclear research facility that sits on 140 acres of land in Penn township, Westmoreland county (Figure 5). The site is currently unoccupied, and ideal for research experiments and realistic demonstrations of the VSAM IFD testbed system, using both ground-based and airborne sensors to cooperatively track vehicles and people moving through an outdoor environment.

Bushy Run is 30 minutes from the CMU campus, and has expansive open spaces, tree lined fields with varying degrees of ground vegetation, and two empty two-story buildings along paved roads. The buildings, roadways, and natural terrain at the site, combined with the facility's limited access to the public, make it an ideal location for controlled experiments and demonstrations involving moving object detection and tracking, as well as for conducting potentially dangerous flight tests involving experimental aerial platforms without endangering human bystanders.

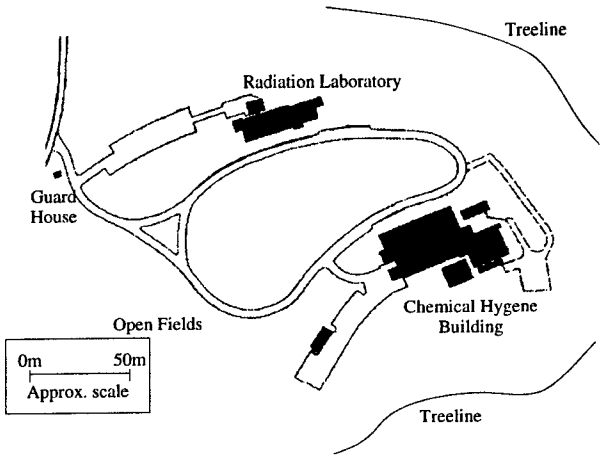


Figure 5: The CMU Bushy Run demo site.

5.2 Year 1 Demonstrations

Two IU capabilities will be highlighted during the first year: cooperative ground-based surveillance, and multi-target tracking by an airborne sensor. Below is a brief description of the objectives of each, coupled with tentative military scenarios that set the stage for the demonstrated IU capabilities.

Cooperative Ground-Based Surveillance

Consider a facility monitoring scenario, which involves continuous surveillance and monitoring of a military facility such as a base or warehouse complex. The site is assumed to be familiar, and detailed site-specific information (e.g. site

models) are available. The site is also assumed to be too large to monitor with a single camera.

Several ground-based stationary sensors are mounted throughout the facility, and along its perimeter. A central operator control unit allows security personnel to analyze information gathered by the sensors. The operator is alerted if vehicles or pedestrians attempt to breach the perimeter in a location other than a normal facility entrance. The system also monitors for suspicious activity within the compound, presenting video clips of interesting events to the operator for review. The operator may designate targets of interest, and the system automatically tracks them through the course of their movements. The key IU capability to be demonstrated occurs as the system hands-off control from one stationary sensor to another, following the target as it enters and exits the fields-of-view of the different sensors. The goal is to maintain a continuous visual lock on the target, as it travels through the compound.

Multi-Target Tracking by a Single Sensor

Consider the need to provide instantaneous situational awareness on the battlefield, where multiple friendly and enemy forces are simultaneously engaged over a large area. A single unmanned air vehicle (UAV) is deployed to circle the battlefield in order to send back timely information on the locations of the combatants. The battlefield is too large to fit in a single field of view when the sensor is focussed at a resolution high enough to distinguish friendly from enemy forces. Nonetheless, it is desired to detect and track as many moving objects as possible, given the limited resources available.

To handle this situation, the UAV VSAM system is instructed to operate in multi-tasking mode, and the sensor begins to scan the scene. As the field of view passes each moving target, it's location is noted with respect to a reference mosaic in which pixel locations are directly related to geographic coordinates (using known transformations calibrated previously). The sensor continously pans and tilts around the scene, noting new targets as they become visibile for the first time. After a quick scan to summarize the positions of moving objects in the scene, the positions of targets of interest are continuously updated by switching the sensor field of view between each of them in turn, using a combination of visual and inertial information to determine where to scan. When returning to update the position of an object, the search begins from its expected new location, given its last known position and trajectory.

6 Evaluation Plan

Key features of the IFD VSAM research program are 1) cooperative use of multiple sensors and 2) moving platforms to provide 3) broad area surveillance and 4) real-time tracking in 5) cluttered and urban environments. We will evaluate the IFD testbed architecture and component IU technologies along several dimensions to measure system competence with respect to each of these features.

False alarm rates for target detection and cueing will be measured with respect to a number of varying factors such as size and distance of the target from the sensor, speed and direction of target trajectory, amount of scene clutter, and number of targets that are simultaneously in view. The sensitivity of moving object detection and tracking processes to ego-motion of the sensor platform will be evaluated for both pan-tilt systems and general vehicular (ground and air) motion. The effectiveness of multi-sensor VSAM integration will be measured by quantifying spatial and temporal discontinuity induced in perceived object trajectories as tracking control is passed between adjacent sensors. We will experimentally determine how large an area can be reliably monitored by a given number of fixed and moving sensor platforms, and how each sensor should be deployed to maximize VSAM performance. The accuracy with which sensor occlusion can be predicted using static scene models and dynamic target models will also be addressed.

The main use of multi-sensor integration in this system is to accurately localize targets within the 3D scene. Geolocations of observed targets will be computed in a number of ways: by multi-image triangulation if the target is

viewed simultaneously by multiple sensors, by range-from-size computations or backprojection of target center of mass onto a collateral terrain map if the target is viewed by a single sensor only, and by extrapolating from the last known trajectory if the target is currently occluded from all sensor viewpoints. In each case, accuracy for the computed target location and trajectory will be evaluated by measuring the deviation between estimated and actual locations of ground truth targets with respect to the number and configuration of sensor platforms.

Beyond these systematic tests of system capabilities, the IFD testbed will also be exercised under a variety of weather conditions and at night (using infrared and laser ranging sensors) in order to assess how these environmental elements and sensor modalities affect system performance.

7 Conclusion

Carnegie Mellon University and the David Sarnoff Research Center are developing a cooperative, multi-sensor video surveillance and monitoring system. Multiple sensors on stationary and moving platforms will cooperate to continuously track moving targets through large, cluttered environments. Extracted target and ephemeris data is collected at an operator control station, and combined with prior collateral information to build and maintain an evolving, dynamic representation of the scene. A single human operator will be able to interact with this scene representation through a graphical user interface, allowing him or her to effectively task the multiple sensors and monitor targets over a large area. An experimental testbed system is being built to support evaluation and demonstration of these and other VSAM technologies being developed within the DARPA IU community.

References

[1] J. Costeira and T. Kanade, "A multi-body factorization method for motion analysis," *Proc. ARPA Image Understanding Workshop,* 1996, pp.1013–1025.

[2] L.S. Davis, R. Bajcsy, M. Herman, and R. Nelson, "RSTA on the move: detection and tracking of moving objects from an autonomous mobile platform," *Proc. ARPA Image Understanding Workshop,* 1996, pp.651–664.

[3] M. Hansen, P. Anandan, G. van der Wal, K. Dana, P. Burt. , "Real-time scene stabilization and mosaic construction," *IEEE Workshop on Applications of Computer Vision,* 1994.

[4] M. Irani and P. Anandan, "A unified approach to moving object detection in 2D and 3D scenes," *Proc. ARPA Image Understanding Workshop,* 1996, pp.707–718.

[5] R. Kumar, H. Sawhney and J. Asmuth, "Geospatial registration," *Proc. DARPA Image Understanding Workshop,* 1997, this proceedings.

[6] R. Kumar, P. Anandan and K. Hanna, "Shape recovery from multiple views: a parallax based approach," *Proc. Darpa Image Understanding Workshop,* 1994.

[7] L. Matthies, R. Szeliski and T. Kanade, "Kalman filter-based algorithms for estimating depth from image sequences," *International Journal of Computer Vision,* Vol.3, 1989.

[8] H. Sawhney and R. Kumar, "True multi-view registration with application to automosaicing and lens distortion correction," *Proc. DARPA Image Understanding Workshop,* 1997, this proceedings.

[9] J. Shi and C. Tomasi, "Good features to track," *IEEE Conference on Computer Vision and Pattern Recognition,* 1994, pp. 593–600.

[10] C. Tomasi and T. Kanade, "Shape and motion from image streams: factorization method," *International Journal of Computer Vision,* Vol. 9(2), 1992.

Video Processing for Security, Surveillance and Monitoring *

P. Anandan and Peter Burt
David Sarnoff Research Center, Princeton, NJ
E-MAIL: {panandan,pburt}@sarnoff.com

Abstract

A major focus of Image Understanding (IU) research activity at the David Sarnoff Research Center is video processing for security, surveillance and monitoring. We have been developing vision technologies for surveillance video processing to enhance battlefield situational awareness, and for a number of Government and commercial vision-based security applications. Our work derives significantly on our strengths in real-time video processing, especially in the areas of video stabilization and mosaicking, registration of video sequences to geocalibrated reference imagery, moving target and intruder detection, multi-sensor image alignment, multi-sensor image fusion, stereo vision, and active multi-resolution pattern recognition and location. This report provides an overview of our research activities in these and other related areas of Image Understanding over the past year.

1 Introduction

The Image Understanding Research (IU) activity at the David Sarnoff Research Center focuses on developing fundamental solutions to real-world applications. We have been developing theory, algorithms, and systems in the area of image and video processing. A major component of our IU research focuses on vision for security, surveillance and monitoring. These include military applications, particularly aimed at video processing to enhance the battlefield situational awareness of the war-fighter, as well as Government and commercial vision-based security applications.

In the area of surveillance video processing, our core technical components include real-time video mosaicking, developing compact representations of the spatial and temporal components of surveillance video information, using these compact representations for efficient video indexing, registering the video sequences and mosaics to calibrated reference imagery, multi-sensor image alignment and fusion Sarnoff's work in the area of security includes vision for physical security (e.g., intruder detection in outdoor warehouse environments), and vision-based acquisition and verification of human iris images.

Recently a number of other applications have emerged as off-shoots of Sarnoff's work in the area of video mosaicking. A particular application with significant commercial potential is a technology called "Video Brush" which enables fast mosaic construction on a PC from video acquired by a hand-held camera.

In addition to these aforementioned areas, Sarnoff has also developed and fielded commercial vision systems for monitoring highway traf-

*The work described here was supported in part by DARPA/ISO under contract No. DAAA15-93-C-0061, the National Information Display Laboratory, US Army Physical Security Equipment Management Office under contract no. DAAK 70-93-C-0066, DARPA/ETO under contract No.DABT63-95-C-0057, NASA Ames Research Center under contract no. NAS2-14301, Sensar Inc., NJ., and PEEK Transyt, Fl.

fic and for traffic signal control. Finally, we have also been conducting basic research in the area of physics based modeling and analysis of fluid flow. This work is aimed at the measurement of flow in a variety of real world fluidic devices, especially, fluidic micro-electrical mechanical systems (MEMS). Last but not least, Sarnoff has also continued the development of advanced architectures and hardware systems for video processing for all of the types of applications described above.

2 Video Processing for Battlefield Situational Awareness

The ease and flexibility of video collection has resulted in the emergence of video as an important source of data for providing battlefield situational awareness. Cameras mounted on airborne platforms, such as the Predator Unmanned Aerial Vehicle or the manned P3 aircrafts are routinely collecting video data over important regions of the world, including Bosnia and Zaire. While the video data collected by these platforms contains a wealth of timely information, it is poorly calibrated, and is hard to use for analytic purposes. Important and interesting data is often buried within the video, hidden among vast amount of irrelevant data. The standard sequential methods of organizing the data, and the standard movie mode of visualization are inadequate for the timely access and processing of that information.

Sarnoff has been developing a comprehensive solution to the problem of video processing for battlefield situational awareness. Our approach is based on transforming video sequence from "frames" to "scenes" using what we refer to as a "Mosaic-based Video Representation" [Irani *et al.*, 1996]. Such a representation is constructed by aligning the frames of the sequence to each other and creating a panoramic view of the static scene, and separating moving objects from the background and representing them as visual events. The individual source frames and the mosaics are registered to calibrated reference imagery in order to fuse the video with other types of geospatial data. The mosaics and the video clips are organized according to time, date, and geolocation. Various modes of access and visualization of the data are provided. Below we briefly review some of the major steps in processing the video.

2.1 Real-time Mosaic Construction

Sarnoff has previously demonstrated the capability for real-time mosaicking of surveillance video using a real-time image processing hardware system called the Vision Front End (VFE-100) processor. This was developed in part under its previous DARPA contract, as part of the DARPA UGV Demo II project. [Hansen *et al.*, 1996]. More recently, we have developed a real-time mosaicking system called the Video Exploitation Workstation Software (ViEWS), under the sponsorship of the National Information Display Laboratory (NIDL). ViEWS consists of a VFE-100 system connected to a Sun Sparcstation. The VFE hardware system performs real-time frame-to-frame alignment at about 8-10 frames/sec. The digital imagery and the alignment parameters are transmitted to the Sun host, which is used to build and store the mosaic image files. A Graphical User Interface (GUI) on the host allows the user to control the alignment and mosaic construction process.

The ViEWS system is currently being used at a number of military locations in Europe in connection with the ongoing US military operations in Bosnia. These systems routinely process video acquired by the Predator UAV. In addition, one system is deployed in Zaire, and processes video acquired by sensors mounted on a manned P3 aircraft.

2.2 Mosaic-based Video Representation

Once the frames are aligned to each other (as in the ViEWS system) the video data is divided into (1) a panoramic mosaic that captures the background scene, (2) the geometric transformation that relates each frame to the mosaic coordinate system, and (3) residual information that captures moving objects and other changes. This representations provides revolutionary ways of accessing frames inside a video:

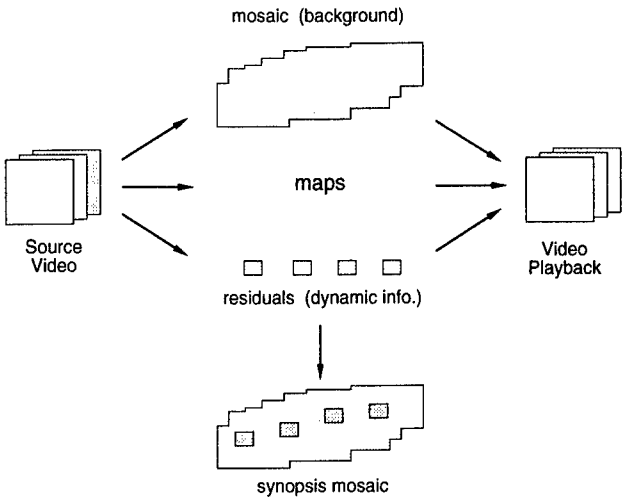


Figure 1: Mosaic-based Video Representation

by pointing to a desired location or a desired object, the relevant video segment can be selected. Other indices such as color/texture features and motion information allow indexing based on appearance of scenes in the imagery and simple image events. A graphical illustration of this approach for representing video surveillance data is provided in Figure 1. The details of the construction of the mosaic-based video representation are described in [Irani *et al.*, 1996]. Figure 2 provides an example of such a representation applied to a real video sequence obtained from the Predator UAV.

2.3 Video Georegistration

Video acquired by surveillance platforms such as the Predator UAV, is typically poorly calibrated. Although GPS and inertial data can be used to derive an approximate "footprint" of the video frames on the ground, their errors can range up to several 100s of meters. In the case of the high resolution sensors (e.g., 1 m/pixel), this translates into misalignment of hundreds of pixels.

We have been developing techniques for automatically registering video and mosaics to calibrated reference imagery (e.g., orthophotos), together with 3D terrain elevation data. This work was done, in part, under the sponsorship of the National Information Display Laboratory (NIDL). The georegistration technique is described in greater detail in [Kumar *et al.*, 1997]. We are currently developing a real-time system for performing georegistration and updating a geospatial database using the registered imagery.

There are two important reasons to attempt to geolocate the video data to greater degree of accuracy. The first is precision targeting, especially of moving ground targets. The second is in order to update geospatial databases, which are used a variety of applications such as mission planning, flight mission pre-rehearsal, and target identification and annotation. A typical example of this latter type of usage arises in the case of mission pre-rehearsal. Currently, the pilots use archival National imagery and associated 3D terrain elevation data to generate synthetic fly-throughs. However, these imagery while being geographically accurate are *dated*; the image information often do not correspond to the current scene details (e.g., due to seasonal differences, changes in the environmental conditions, and changes in cultural features). Georegistered video offers an ideal alternative for these applications.

2.4 Multi-sensor image alignment and fusion

In order enhance target detection, especially under conditions of poor visibility (e.g, at night), most surveillance platform contain multiple sen-

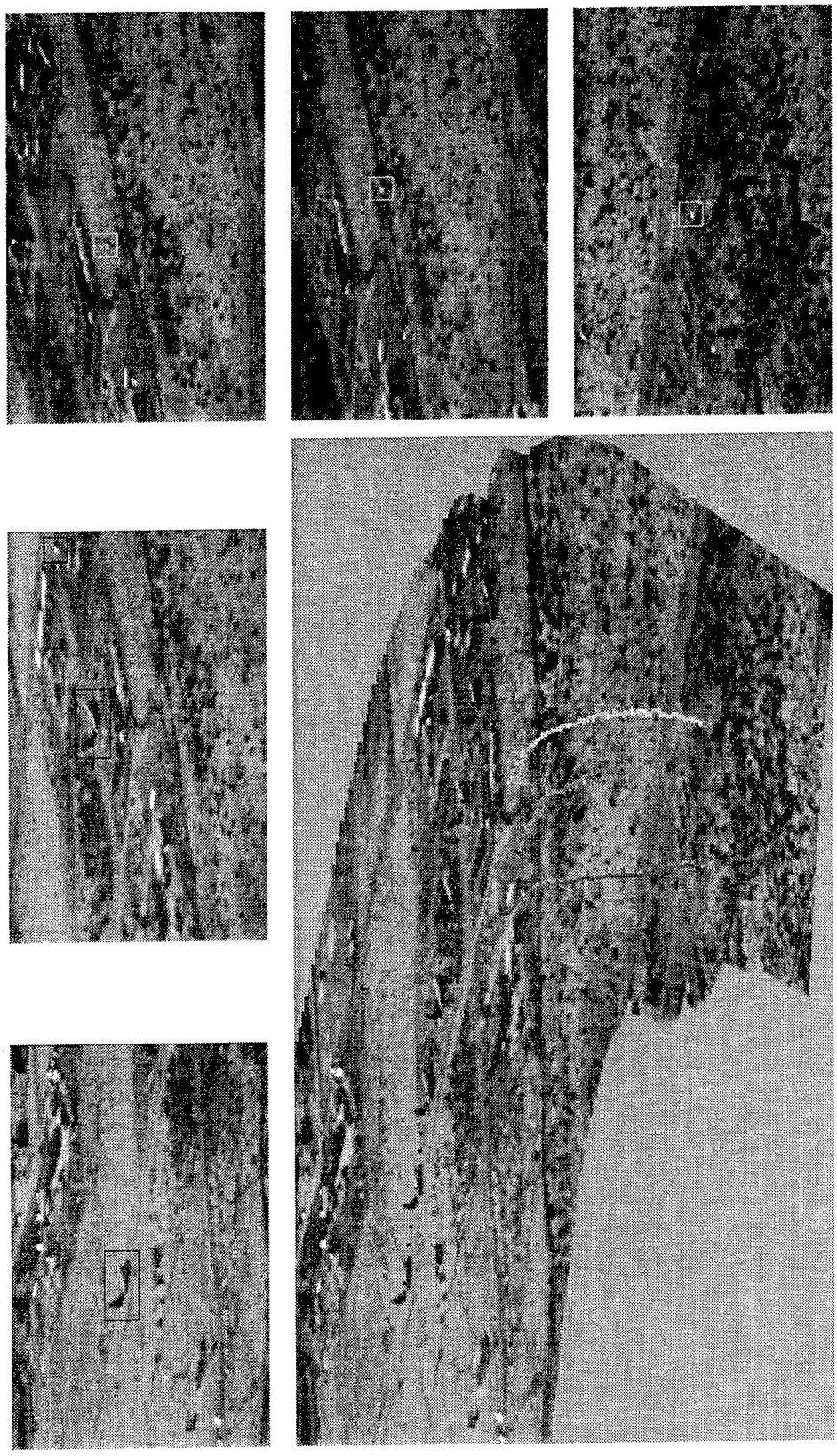


Figure 2: An example of mosaic-based video representation. Sample frames from the input video sequence are shown surrounding the synopsis mosaic constructed from that sequence, which is shown in the middle. Visual and symbolic depictions of the trajectories of an airplane and three parachutes are also shown on the mosaic image.

sors of different sensor modalities. The full exploitation of the information provided by such a sensor suite requires aligning these multi-sensor images. Also, often these sensors capture complimentary types of visual information. A single fused display of the visual imagery is useful to provide a human operator with a clear and immediate sense of the observed data.

Under a NASA sponsored project, Sarnoff has developed a technique for registering multi-sensor images that performs global parametric alignment using a *locally* invariant match measure. Our approach is based on the observation that the physical structure of the scene gives rise to correlated information in the images, but such information is only *locally* correlated. Also, in order to recover large displacements between the two images, the overall estimation process is couched within a coarse-to-fine refinement strategy using multi-resolution image information. However, in order to capture the distribution of the physical structures over various spatial scales, a special type of multi-resolution representation known as the *integrated feature pyramid* is used instead of the standard pyramid representations, such as Gaussian or Laplacian pyramids and other wavelet based representations. This work is described in [Irani and Anandan, 1997].

Sarnoff has also made a significant advance in its fusion algorithms, under the same NASA project. Sarnoff had previously introduced a technique called "pattern selective image fusion" that is implemented using pyramid/wavelet image transform. This has proven effective in a remarkable range of applications, from combining IR and visible surveillance imagery to combining CT and MR medical imagery. Sarnoff built real time hardware to perform image alignment and fusion several years ago. During the past year Sarnoff undertook a detailed analysis of sampling and aliasing in pyramid/wavelet transforms. This has led to important advances in multiresolution analysis that both improve and simplify the fusion algorithms. These advances should improved performance of many other pyramid/wavelet based image processing functions as well, such as motion detection and stereo.

3 Vision for Security Applications

Sarnoff's work in the area of security focuses both on Government applications and Commercial security applications. Our Government sponsored work is aimed towards physical security, whereas the commercial application focuses on non-intrusive human iris acquisition and verification.

As part of the Exterior Mobile Assessment, Detection and Response System (MDARS-E) program sponsored by the US Army Physical Security Equipment Management Office, Sarnoff has developed a system for the real-time detection of intruders a a large outdoor storage facility. Our approach is based on making a comparison of the "current" visual image of the scene with a previously stored "reference" image containing only the background scene. Irrelevant changes due to noise, image misalignments, brightness changes, wind-blow clouds and vegetation are eliminated using a variety of spatio-temporal image analysis techniques. In order to be adaptive to slowly varying daylight, weather, and illumination conditions, the reference image is continually updated. The result is a system that achieves high probability of detection of targets such as vehicles, humans, and animals, while maintaining a low false-alarm rate.

Sarnoff's work on iris acquisition and verification was developed under the sponsorship of Sensar, Inc. This system uses active vision techniques for the acquisition of the iris images. The user simply stands in front of the system, an image of their iris is acquired, and their identity is verified or refuted. The user is not required to make physical contact with the system or to assume any particular pose except that he stand with his head within a designated calibrated volume. This system consists of a stereo pair of wide field-of-view cameras (WFOV), a narrow field-of-view camera (NFOV), and a pan-tilt mirror allowing the NFOV to be moved relative to the WFOV (see Figure 3. Stereo analysis of the WFOV camera images is used to detect locate the person's head within the 3D calibrated volume, and template methods are used to precisely locate the right eye. This information is fed to the NFOV camera system, which is

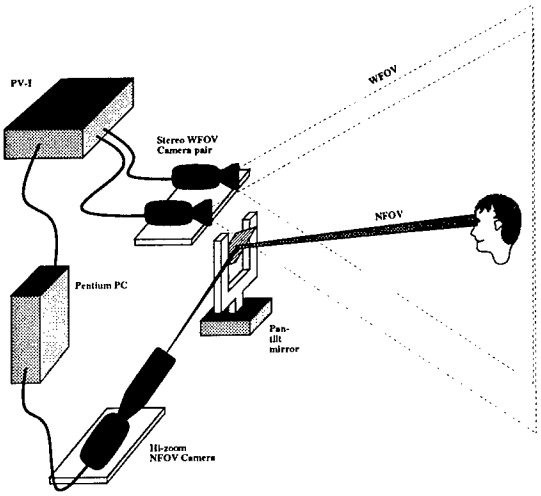


Figure 3: The overall configuration of the IRIS recognition system.

actively controlled to obtain a close-up view of the eye, and the high resolution image obtained by that camera is fed to Daugman's IRISCAN system [Daugman, 1993] to verify or refute the identity of the user. An typical example of an input image captured by the WFOV camera is shown in Figure 4. A typical example of the output provided by the NFOV camera after localizing the eye is shown in Figure 5. For a detailed description of this work, see [Hanna *et al.*, 1996].

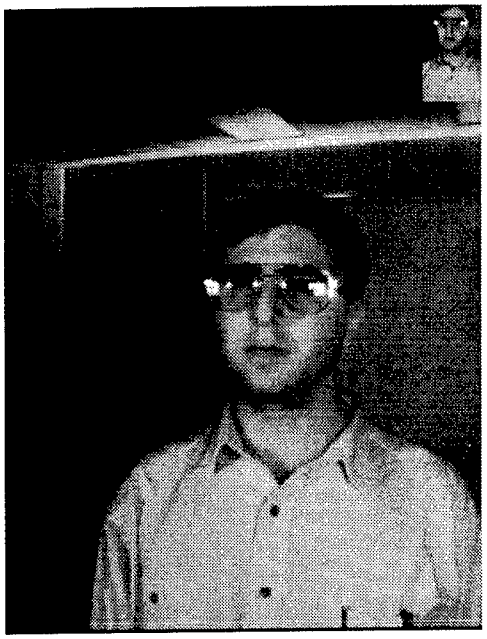


Figure 4: Iris system input. A typical example of the input image to the system captured by the WFOV camera. The inset in the upper right corner shows at reduced resolution the area selected by the head-finding stereo system.

4 Other Ongoing IU Activity

This section briefly summarizes several other components of Sarnoff's IU research, besides the security, surveillance, and monitoring activities that were described above.

Panoramic Mosaics with VideoBrush: As an offshoot of our work on image alignment and video mosaicking, we have developed a system called the "VideoBrush", which allows fast mosaicking on a PC without any special purposed hardware. This system uses an idea called "Manifold Projection", which refers to the sweeping of the scene using a one dimensional sensor array. The key steps in this process are 2D frame-to-frame alignment to compensate for image plane translations and rotations, cutting and assembling central strips

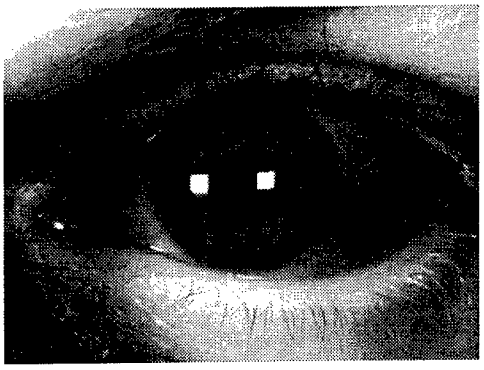


Figure 5: Iris system NFOV output. A typical example of the output image provided by the NFOV system after localizing the eye.

from the aligned images to form a mosaic image, and blending using a multi-resolution spline approach to remove seams. This system is implemented on a Pentium PC platform and can be used to make color mosaic images using a hand-held camera. A convenient interface allows the user to preview the mosaicking process in real-time as the hand-held camera is used to scan the scene, and then to construct a color mosaic image within a few minutes. More details on this work can be found in [Peleg and Herman, 1997].

In order to allow the greatest flexibility in the use of our registration techniques, we make no assumptions about the camera lenses or camera calibration. Often the camera focal length, and the image center may be unknown, and the camera lens will often introduce spatial distortions to the image. When multiple images are aligned to form a single mosaic, it is also desirable to visualize the mosaic in a coordinate system which is free of distortion. This means that the reference coordinate system must be different from that of any of the input images. To meet these requirements, we have developed a registration technique which automatically corrects for lens distortion and automatically selects a reference coordinate system for the mosaic, which is free of distortions. This work is described in greater detail in [Sawhney and Kumar, 1997].

Traffic Monitoring: Under the sponsorship of PEEK Transyt, Sarnoff has been developing a system called the PVS traffic monitoring system for use on highways and at road intersections. The highway application involves collecting vehicle statistics such as number of cars per a given time, and the distances between the cars. The intersection application involves detecting the presence of vehicles at a traffic intersection in order to control the operation of the signal lights. Both systems operate in real-time (30Hz), with the ability to multi-task between upto 4 cameras. The system consists of three double-sided 3U VME boards and is compact enough for field operation, and can be manufactured at minimal cost. The PVS system operates by maintaining an evolving reference image of the scene as it would appear if no vehicles were present, and by comparing incoming image frames to the reference frames in order to detect vehicle presence. Details of the system can be found in [Wixson, 1996]. Over 200 systems have been fielded to date in various parts of the US and in Europe.

Physics based modeling of fluid flow: Under the sponsorship of DARPA/ETO, Sarnoff has been addressing the problem of recovering quantitative measurements of fluid flow from corresponding image sequences. Sarnoff's work is aimed at augmenting the arsenal of tools that are available for the measurement of flow in experimental fluid mechanics using visual analysis of the motion of tracer particles. We are particularly motivated by applications in microfluidics; however, the methods that we have developed are equally applicable to macroscopic flows. Based on physical principles derived from fluid mechanics, we have developed a novel motion recovery algorithm. This has led to two classes of constraints on imaged flows. The first class of constraints are differential constraints that arise from conservation principles (e.g., conservation of mass, conservation of momentum) as applied to fluid mechanics. The second class of constraints come about as boundary conditions on the permissible flows. These constraints have been further bolstered by consideration of smoothness constraints on the flow that serve to ameliorate the effects of noise. These constraints are combined using a calculus-of-variations formulation to yield a pair of partial differential equations that relate measurements of image intensity to flow field components. A numerical solution has been derived via discretization; a corresponding algorithm has been instantiated in software. This implementation has been evaluated using synthetic and natural image sequences that depict fluid flow. For the synthetic imagery as well as natural imagery where the expected flow can be predicted analytically, the recovered flow has shown very small root mean square error. Currently, we are applying the algorithm to the measurement of flow in a variety of real world fluidic devices, especially, fluidic micro-electrical mechanical sys-

tems (MEMS). This work is described in greater detail in [Wildes *et al.*, 1997], which is also published in these proceedings.

Real-time Hardware: A significant component of Sarnoff's vision research program has been in the development of advanced architectures and hardware for real time vision applications. Sarnoff has developed a sequence of processing chips and platforms for research and commercial products. Over the past year, Sarnoff has completed a second generation pyramid image processing chip that runs at 60 MHz, and has completed commercial processing modules for traffic monitoring and iris recognition. Sarnoff's current development is focused on the VFE 200 (vision front end) system, to be completed in mid 1997. This development is supported by the Army MDARS program and will result in a highly modular and flexible family of processing boards designed to support real time autonomous driving and surveillance applications.

5 Conclusion

During the past year Sarnoff has continued its advanced research and development of IU technologies. A major focus of effort has been vision technologies for security, surveillance, and monitoring. In addition, we have also continued our work in other areas of Image Understanding. This report provided an overview of Sarnoff's IU activities during the past year.

References

[Daugman, 1993] J. Daugman. High confidence visual recognition of persons by a test of statistical independence. 15(11):1148–1160, 1993.

[Hanna *et al.*, 1996] K. Hanna, R. Mandelbaum, D. Mishra, V. Paragano, and L. Wixson. A system for non-intrusive human iris acquisition and identification. In *International Association of Pattern Recognition Workshop on Machine Vision Applications*, 1996.

[Hansen *et al.*, 1996] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt. Real-time scene stabilization and mosaic construction. In *ARPA Image Understanding Workshop*, pages 457–465, Monterey, CA, November 1996.

[Irani and Anandan, 1997] M. Irani and P. Anandan. Robust multi-sensor image alignment. In *DARPA Image Understanding Workshop*, New Orleans, LA, May 1997.

[Irani *et al.*, 1996] M. Irani, P. Anandan, Jim Bergen, R. Kumar, and S. Hsu. Efficient representations of video sequences and their applications. *Signal Processing: Image Communication*, 8:327–351, 1996.

[Kumar *et al.*, 1997] R. Kumar, H. Sawhney, and J. Asmuth. Geospatial registration. In *DARPA Image Understanding Workshop*, New Orleans, LA, May 1997.

[Peleg and Herman, 1997] S. Peleg and J. Herman. Panoramic mosaics with videobrush. In *RPA Image Understanding Workshop*, New Orleans, LA, May 1997.

[Sawhney and Kumar, 1997] H. Sawhney and R. Kumar. Multi-image alignment. In *DARPA Image Understanding Workshop*, New Orleans, LA, May 1997.

[Wildes *et al.*, 1997] R P Wildes, M J Amabile, A M Lanzillotto, and T S Leu. Experiments with an algorithm for recovering fluid flow from video imagery. In *DARPA Image Understanding Workshop*, New Orleans, LA, 1997.

[Wixson, 1996] L. Wixson. Illumination assessment in a video-based traffic monitoring system. 1996.

Visual Surveillance and Monitoring of Human and Vehicular Activity

Larry Davis Rama Chellappa
Yaser Yacoob Qinfen Zheng
Center for Automation Research, University of Maryland
College Park, MD 20742-3275 (lsd@umiacs.umd.edu)

Abstract

This report describes research on visual surveillance of human and vehicular activity in urban battlefield sites to be conducted under the VSAM component of DARPA's Image Understanding Program. We first give a brief introduction to our research program, emphasizing its goals, technical approaches and relevance to battlefield awareness, and then illustrate the approaches we are pursuing through three examples. The first involves a low-level detection and tracking system that can, in real time, track people and their parts through video imagery; the second is a high-level system for recognition of multi-person actions and goal-directed control of low- and intermediate-level vision components; and the third is a real-time image stabilization algorithm to support surveillance from a moving platform.

1 Introduction

We are exploring fundamental research problems related to the analysis of visual (monochromatic video and IR) sensory sources for visual surveillance of human and vehicular activity in urban areas for military or law enforcement purposes.

Our vision of an autonomous urban battlefield surveillance system, illustrated in Figure 1, involves a distributed suite of heterogeneous and relocatable sensors—in our proposed research, infrared and video cameras—monitoring a large geographic area, in the context of a site model, for the entrances, exits, and activities of people and vehicles. The site model contains knowledge used by the surveillance system to focus its attention and constrain its image analysis to detect people, vehicles and their interactions. The site model could include representations for buildings, roadways and structures such as lamp-posts which might occlude actions being observed from different vantage points. Scarce human operators must monitor the outputs of the surveillance system, under constraints of limited bandwidth and possibly severe psychological pressure.

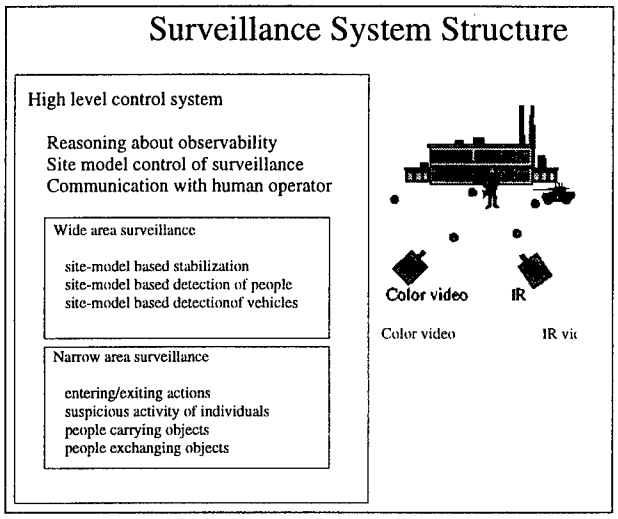


Figure 1: Organization of surveillance system

We specifically envision a surveillance system for monitoring the urban battlefield, where the movements and actions of even a small number of individuals and a small amount of equipment can lead to a great loss of life, and in which one must rely on incomplete and qualitative site modeling to control and focus perception systems. It is critical that such battlefields be monitored for snipers, introduction of offensive weapons such as hand-held missile systems, formation of crowds that might lead to riots, movements of transport vehicles, etc.

It is the responsibility of the surveillance system to

This project will be supported by the Defense Advanced Research Projects Agency (ARPA Order No. E653) and the U.S. Army Research Laboratory. For further information see
http://www.umiacs.umd.edu/users/lsd/vsam.html.

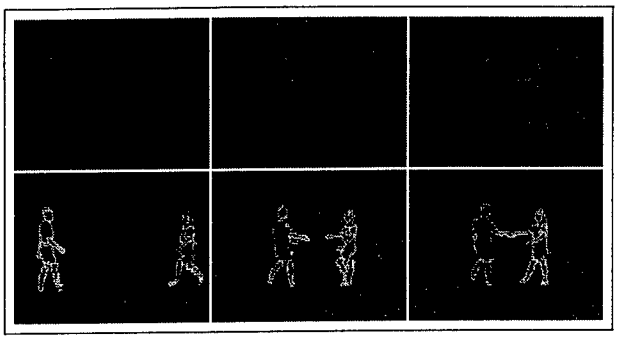


Figure 2: IR handshaking sequence

screen the wide area being monitored, and to communicate with a human expert when situations arise that demand his or her attention (say to allocate resources for additional surveillance, or to make decisions that would lead to military action). In order for the surveillance system to monitor an area of any significant geographic extent, it must employ a suite of sensor platforms which, generally, must be relocatable to bring more surveillance power to bear on potentially interesting situations or simply to provide adequate coverage of a large surveillance site. However, to control the cost and complexity of the surveillance system, the number of sensor platforms must be limited. This suggests that the surveillance system should employ multiple levels of analysis of the area. In our project we consider two levels—a coarse level (wide-area surveillance) in which a significant portion of the area is monitored at low resolution, and a fine level (narrow area surveillance) in which a much smaller area is monitored either at much higher resolution, or with much more detailed analysis of and reasoning about the movements and actions of people and vehicles.

"Wide-area" surveillance employs sensors with large fields of view to detect potential activities of interest, invoking "narrow-angle" surveillance subsystems that perform more detailed analyses of human and vehicle motion to recognize specific activities requiring operator intervention. Wide-area site surveillance detects, in real time, when vehicles and people enter the surveillance area, and must track their motions while in the surveillance area. Figure 2 shows three frames of an IR sequence in which two people approach one another and shake hands. The results of a segmentation-based motion detection algorithm is outlined on the original IR imagery. This algorithm, which can process up to 15 frames per second of IR imagery on a dual processor Pentium, is discussed in Section 2.

Narrow-area surveillance will be focused on person-vehicle interactions (people entering/ leaving vehicles, placing objects under vehicles) and person-object-person interactions (picking up, putting down, carrying and exchanging of objects). Figure 3 shows several frames in a video sequence in which one person approaches a second, takes an object away from the second person, and then runs off, an example of a "mugging" action. In Section 3 we explain how this type of action can be recognized by a high-level system that maintains and applies temporal activity models to video sequences.

Since, in many real surveillance scenarios, relocatable sensors must be moved to obtain adequate site coverage, our research will also consider problems related to the analysis and integration of visual information from a moving sensor. In particular, we will develop new algorithms for image stabilization using 3-D site model information, and for detection of independently moving people and objects using site-model-based video data processing. Section 4 reviews some of our prior work on real-time algorithms for image stabilization and discusses our research plans on 3-D site-model-based image stabilization.

2 Detection and tracking of humans and vehicles

Here, we briefly discuss preliminary results obtained on the development of real-time algorithms for detecting and tracking of moving objects, specifically people and vehicles, from a stationary sensor. A variety of algorithms have been developed recently for detecting people (and their parts) and tracking them through time. For example, the Pfinder system [1] from M.I.T.'s Media Laboratory uses color stereo cameras to track a person's head and hands in real time. Other systems - for example, [9] and [16] - also employ color, adapting prior models for human skin to detect and track faces in real time.

In outdoor and low-light conditions, these strong color cues will not be available to support detection and tracking. Instead, cues based on how objects move, where they occur in the context of available site information, and their shapes and sizes will have to be employed for both detection and tracking. Additionally, in outdoor environments there are many other sources of "nuisance" motion, including small-magnitude motions due to vegetation and the presence of small ground and air animals whose appearances and motions are probably not known to the vision system.

A common approach to moving object detection involves a combination of background subtraction, image morphology to remove isolated noise detections and connect the inevitable fragments of objects produced by the pixel-based detection decisions, and then (optionally) feature extraction applied to the

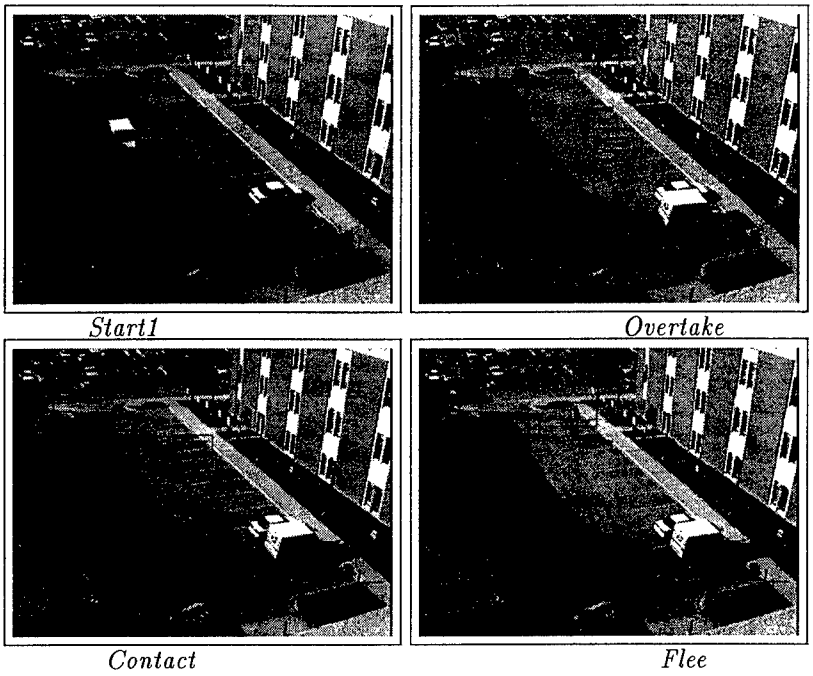


Figure 3: Four key frames in the recognition of a "mugging" action

connected components of the resulting binary image. A good example of such a system is [6]. A potential drawback of applying this approach to outdoor and low-light scenes is that the morphology produces highly inaccurate representations of the moving objects - one must ordinarily employ several dilation steps to connect the fragments created by the noise-reducing erosion and the unreliable pixel-based background subtraction.

As an alternative, we have been studying the use of a segmentation-based approach to moving object detection. The basic steps involved are:

1. Creation of a background model, similar to the method employed in the Pfinder system, in which we associate a gray-scale distribution with each image pixel based on several seconds of observations. Robust estimation of this distribution allows us to overcome the effects of moving objects in the scene.
2. Pixel-based background subtraction, followed by morphological operations to remove small noise components.
3. Regions containing the foreground components are then segmented using a hierarchical segmentation algorithm, described in [11]. A segment is assigned to the foreground if a sufficient percentage of its area is composed of foreground pixels from the preceding step.

Basing the detection of foreground objects on the segmentation results in more accurate delineation of moving objects; more importantly, the hierarchical graph structure describing the connected sets of foreground regions can be employed to track the foreground objects.

This algorithm has been implemented on a dual-processor Pentium PC and can process up to 15 frames per second (each frame containing approximately 160 × 120 pixels) of either IR imagery or subsampled monochromatic video.

3 Action recognition

Most prior research on recognition of human actions has focused on gestures and other stylized actions, for which techniques such as hidden Markov models are appropriate. Examples include reading ASL (Starner and Pentland [15]), and our own work on head gesture recognition (Morimoto, Yacoob and Davis [8]). Researchers have also attempted to take advantage of the periodicity of continuous human motions like walking, running, etc. to recognize them, either from preferred viewing directions, as in Rohr [12], or from general perspectives using invariant signatures (Seitz and Dyer [13]). Campbell and Bobick [2] use phase space techniques (related to work in [14]) to recognize dance steps.

Our research, in contrast, is focused on the representation and recognition of less stylized interactions among people and vehicles, and will make strong use of site knowledge to constrain and focus image analysis algorithms.

The system we are constructing builds on ideas de-

veloped in [3], [4], [5], and [10]. It uses logic programming to represent and apply temporal logic programs to the analysis of surveillance video. The logic programs, through the use of a site model, control where and how to apply image analysis algorithms to detect and track people and vehicles.

An example is provided in Figure 3. Here we show four frames from a 30-second sequence in which one person "mugs" another. A temporal logic program defines a mugging activity as one that includes three intervals, corresponding intuitively to the mugger overtaking the victim, coming into physical contact with the victim, and fleeing. The system knows that the action is taking place on a ground plane, roughly calibrated in the frame of the camera. The ground plane model is used to scale templates of the participants of the mugging (initialized when the participants enter the field of view) as they move away from the camera, and also to estimate the 3-D velocities of the participants, and whether they are close enough in the world to allow physical contact. Sun angle information is used to predict rough shadow size for a person of nominal height, and also the direction in which the shadow is cast on the ground plane. Shadows can be (and in this example are) used to assist in the tracking of a person or a vehicle. The four frames in the Figure show the initialization of the tracking, and representative frames during the detected intervals of overtaking, contact and fleeing.

4 Site-model based stabilization

Image stabilization is the process of generating a compensated video sequence where unwanted motion of the camera is removed from the original input. It can be used as a front-end system for many tasks that require dynamic image analysis, such as scene change detection, tracking of independently moving objects, surveillance, and monitoring. We have implemented [7] a fast and robust electronic digital image stabilization system that can handle large image displacements based on a multi-resolution motion estimation technique. The method tracks a small set of features and fits a similarity or affine motion model to the feature displacements using least-squares. Stabilization is achieved by combining all motion from a reference frame and warping the current frame back to the reference. The system was implemented in a real-time parallel pipeline image processing platform (a Datacube MaxVideo 200 board connected to a SUN SPARCstation 20/612 via a VME bus adaptor), and is able to stabilize images of resolution 128×128×8 bits at approximately 10 frames/second.

Figure 4 shows stabilization results for an outdoor sequence with very large panning and zooming, using the similarity model. The top-left and top-right images show an arbitrary input frame f_i and the corresponding stabilized frame respectively. The bottom-left image shows the difference image between frames f_{i-1} and f_i. The bottom-right image shows the difference between f_{i-1} and f_i stabilized to f_{i-1}. Since after stabilization f_i is aligned to f_{i-1}, the residue is minimized.

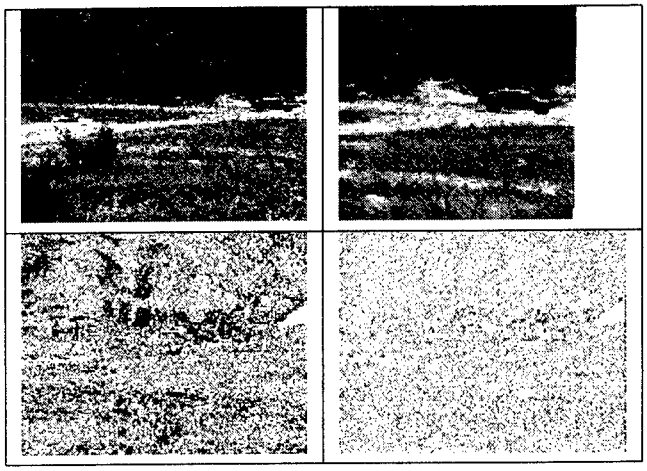


Figure 4: Stabilization results

For better visualization, the motion estimates can be used to align the input image frames and compose a panoramic view of the scene, also known as mosaicking. An example of a mosaic image is shown in Figure 5. This mosaic is composed of 90 frames from the same panning/zooming sequence, which starts zooming out and panning from right to left. The zoom ends after approximately 30 frames. The top image of Figure 5 shows the mosaic after 90 frames, and the bottom row shows frames 90, 45 and 1. Observe the scale change between frames 1 and 45, and how they appear in the mosaic.

References

[1] A. Azerbayejani, C. Wren and S. Pentland, "Real time 3D tracking of the human body," *ImageCom*, 1996, 19-24.

[2] L.W. Campbell and A.F. Bobick, "Recognition of human body motion using phase space constraints," *ICCV*, 1995, 624-630.

[3] S. Dance, T. Caelli and Z.-Q. Liu, "A concurrent, hierarchical approach to symbolic dynamic scene interpretation," *Pattern Recognition*, **29**, 1996, 1891-1904.

[4] D. Harwood, M. Subbarao, H. Hakalahti and L. Davis, "A new class of edge-preserving smoothing filters," *Pattern Recognition Letters*, **6**, 1987, 155-162

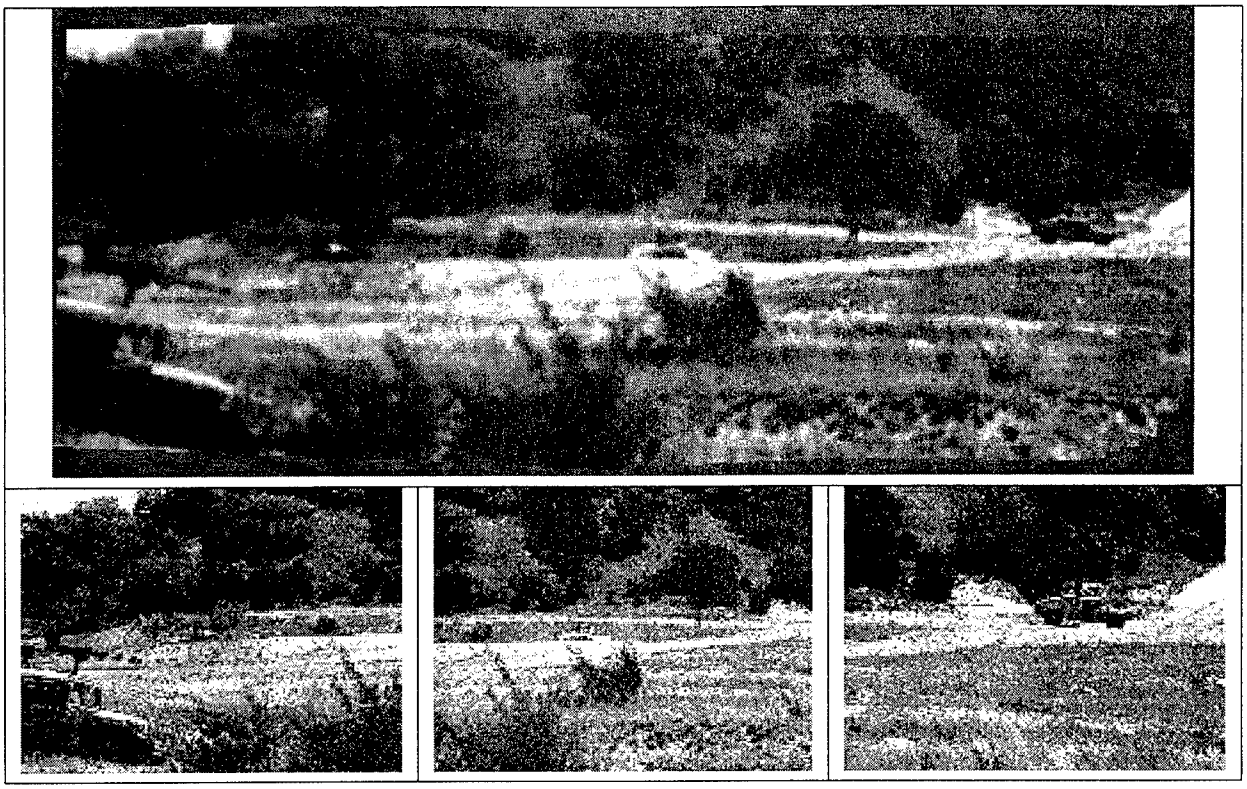


Figure 5: Results of mosaic construction

[5] R. Howarth and H. Buxton, "Visual surveillance monitoring and watching," *ECCV*, 1996, Volume 2, 321-334.

[6] S. Intille, J. Davis and A. Bobick, "Real-time closed world tracking," M.I.T. Media Laboratory Perceptual Computing Section TR 403, 1996.

[7] C. Morimoto and R. Chellappa, "Fast electronic digital image stabilization for off-road navigation," *Real-Time Imaging,* **2**, 1996, 285-296.

[8] C. Morimoto, Y. Yacoob and L.S. Davis, "Recognition of head gestures using hidden Markov models," *ICPR,* 1996, Volume 3, 461-465.

[9] N. Oliver, S. Pentland, F. Berard and J. Coutaz, "LAFTER: Lips and face tracker," M.I.T. Media Laboratory Perceptual Computing Section TR 396, 1996.

[10] C. Pinhanez and A. Bobick, "Approximate world models: Incorporating qualitative and linguistics information into vision systems," *AAAI*, 1996, 1116-1123.

[11] C. Rodriguez, D. Harwood and L. Davis, "An appearance-based approach to object recognition in aerial images," University of Maryland Center for Automation Research TR 746, 1994.

[12] K. Rohr, "Towards model-based recognition of human movements in image sequences," *CVGIP: Image Understanding,* **59**, 1994, 94-115.

[13] S.M. Seitz and C.R. Dyer. Affine invariant detection of periodic motion, *IEEE CVPR*, 1994, 970-975.

[14] E. Shavit and A. Jepson, "Motion understanding using phase portraits," *Looking at People Workshop, IJCAI*, 1993.

[15] T. Starner and A. Pentland, "Visual recognition of American Sign Language using hidden Markov models," *Workshop on Automatic Face and Gesture Recognition,* 1995, 189-194.

[16] J. Yang and A. Waibel, "A real-time face tracker," Carnegie-Mellon University School of Computer Science report, 1996.

VSAM at the MIT Media Laboratory and CBCL: Learning and Understanding Action in Video Imagery*

Aaron Bobick Alex Pentland
MIT Media Laboratory
Cambridge, MA 02139
(bobick, sandy@media.mit.edu)

Tommy Poggio
MIT CBCL
Cambridge, MA 02139
(tp@ai.mit.edu)

Abstract

This report details some initial results and future directions of our research into the machine perception of action. We note that motion understanding can reflect three levels of interpretation — movement, activity, and action. We divide our work into these three areas and also list relevant applications addressed by the respective technologies.

1 Introduction: Action in Video

The recent shift from the processing of static images to the manipulation of video sequences is causing a profound change in the expectations of information consumers. In particular, the ability to exploit "moving" images has generated demand for the direct distillation of descriptions as to what is *happening* in a given situation. Previously, with only static images available, the assumption was that image processing or computer vision could only possibly characterize what was (or was not) in a picture; the assertion as to what was happening in a scene was considered a secondary, higher level operation. Now, however, *actions* may be directly observed. From an information consumer perspective the labeling of a particular space-time region of a video sequence as a "tank platoon re-fueling" should be no different than labeling a region of a static image as a "parking lot."

From the perspective of computer vision, however, the two labeling tasks are fundamentally quite distinct. Whereas we have numerous methods for representing both image and geometric-scene properties, we have many fewer tools for consideration of action. The goal of this project is to develop new representations of action and of the appearance of action, and embed those representations within appropriate recognition paradigms.

The need for such technology on the part of defense and intelligence agencies is clear. An ever increasing amount of video imagery available along with the concurrent pressure to perform tasks with fewer human assets demands the (semi-) automation of surveillance tasks. Monitoring a room to distinguish between normal versus unusual activities, monitoring a port to determine any unusual actions on the part of a fleet, tracking a non-cooperative entity in a battlefield environment — all these tasks require powerful and flexible representations of action. While the achievement of these tasks still requires extensive basic research, the increased computational capabilities of machines and devices makes it feasible to begin developing the necessary core technologies.

As our project is only just beginning, we have structured this paper to outline the broad technological areas that will be considered, and to place them in context with respect to the applications and scenarios to which they are relevant. For each of the technical sections — Human Movement, Activity Understanding, Context-sensitive Action Recognition — we will briefly mention the recent technical achievements, preview the next research steps to be taken, and enumerate the defense and intelligence applications that will be impacted by the technology.

2 Movement, activity, and action

As work has begun on interpreting video sequences, confusion has arisen over exactly what constitutes "action recognition." From our own research understanding action has ranged from the simple recognition of gross body motions [Bobick and Davis, 1996b] to the interpretation of American Sign Language [Starner and Pentland, 1995] to an automated camera system to monitor action in a highly constrained environment [Bobick and Pinhanez, 1997]. Each of these technologies considers action understanding at a fundamentally different level.

To characterize the different approaches it is useful to construct a taxonomy of motion understanding problems. Each paradigm differ along two fundamental dimensions: time and knowledge. Following the taxonomy recently introduced by Bobick [1997] we define the three levels as *movement*, *activity*, and *action*. Movement refers to simple, atomic motion: opening a door, executing a particular assem-

*This work is sponsored in part by ORD contract 94-F133400-000 and DARPA contract N000-14-95-10521. Home page for this project is at http://vismod.www.media.mit.edu /darpa-vsam

bly step, throwing an object. Because the moves are atomic, manipulating time reduces to only a simple variation in speed. Activities include those temporal events that require sequences of steps, often themselves movements. Examples include gestures or simple behaviors, such as making a left turn in an automobile. The understanding of time is more complex, often requiring significant time warping. Finally, actions refer to activity placed in context, and often making semantic or causal reference.

In the sections that follow we divide our efforts into these three areas of understanding motion, and we list relevant applications.

3 Human movement

Our work in the perception of human movement has focused on detection, tracking, and recognition, with the latter recently focused primarily on appearance-based techniques.

3.1 Detection of people

Although the goal of this project is the understanding of action, an important problem is the detection of people in imagery. We have recently developed a trainable system for detecting people in static imagery. The method is based on a wavelet representation of the image and the definition of an object class in terms of constraints on a subset of the wavelet coefficients [Oren, et. al., 1997]. It is invariant to changes in color and texture, and can be used to robustly define a rich and complex calls of objects such as people.

3.2 Body tracking

We have developed technologies for both the initial acquisition and subsequent tracking of significant body features. Pfinder [Wren, et. al., 1995] is a system that uses a multi-class statistical model of color and shape to segment a person from the background scene, and then to find and track people's heads and hands in a wide variety of viewing conditions. With either multiple cameras or a structured environment, three-dimensional information can be provided as well. Recently, [Azarbayejani and Pentland, 1996] we incorporated stereo imagery into the real-time 3-dimensional tracking of skin-colored hand regions.

3.3 Appearance-based movement recognition

If one takes a video sequence of someone performing an movement (say, sitting down) and blurs it mercilessly (each frame to 15 by 20 pixels) the movement is still immediately apparent when the frames are put in motion.[1] This is the case even though *there are no discernible features in each individual frame.* This simple demonstration indicates that geometric modeling is not necessary to recognize action. And given the difficulties present in computing the 3D structure, it might not even be desirable.

Recently, we have begun to develop appearance-based methods of recognizing movement. The basic idea is to separate where motion is happening in the image (i.e. the shape of the motion field) from how the motion is moving (i.e. the movement of the motion field). By performing temporal integration over simple image differences we create a *temporal template* [Bobick and Davis, 1996b]. Statistical moments describing these templates are used as an index into stored models of movements. The procedure is fast, amenable to multi-camera input, and robust.

An alternative appearance-based approach to dynamical models involves a linear combination of prototype images, each easily "learned." The approach is at least partially motivated by biological insights in how the human visual system recognizes motions and how neurons in IT cortex code for object views and sequences of views. Our past work has focused on the modeling of static objects. A deformable model is created for a class of objects as the linear combination of prototype images and their affine deformations. An object is said to be a member of some class if the deformation from a base example is consistent with examples of that class. This new type of hierarchical flexible model can be used as a generative model to synthesize novel images of the same class. This model can also be used for image analysis by fitting the model parameters to an image via an optimization procedure (Jones and Poggio 1995, Beymer and Poggio 1996).

A particular extension of linear combinations of prototypes appears eminently suitable for use in the domain of dynamic 3D objects such as humans engaged in different activities. Recent psychophysical experiments – extending the famous Johansen's results – strongly suggest that the 2D traces of the 3D trajectories of dynamic objects contain almost all of the perceptually important information for recognition purposes. These results suggest that we can represent motion sequences of (possibly non-rigid) 3D objects as 2D images of the trajectories of a few of their salient feature points. These "trajectory images" (T-images) can then be manipulated in exactly the same fashion as 2D images of static objects. In a learning by examples setting, a few of the T-images can be acquired as the model prototypes. Their linear combinations can then be used to account for a novel T-image. The novel T-image is recognized as belonging to that class of prototype T-images that best explain it.

3.4 Applications of movement recognition

Applications of movement recognition technologies include:

- **Perimeter monitoring.** If an area has deemed clear of personnel, one would like to be able to monitor for intrusion. To avoid falsely detecting the motion of stray objects

[1] See demonstration at http://www-white.media.mit.edu/ jdavis/Actions.action.html

(wind-blown or animals) a movement recognition system could detect walking and other suspect movements.

- **Security surveillance.** Movement recognition is capable of distinguishing between allowed motions (e.g. placing letters in an "inbox") and prohibited activities (e.g. opening file cabinets).

4 Activity parsing

Activities involve sequences and their recognition requires a more comprehensive manipulation of time than interpreting movements. Much of our work in recognizing activities has considered gestures but has recently extended to temporal behaviors such as driving.

4.1 Gesture recognition

Our initial work on gesture recognition modeled gestures as a sequence of explicit states in some feature space. The states are defined in training data, and testing reduces to a dynamic programming search to find the most similar gesture. The method employed the idea of a prototype and thus learned more quickly than many statistically oriented methods [Bobick and Wilson, 1995].

The natural progression from this work was to explore the use of Hidden Markov Models for describing gesture. One innovative technique we introduced allowed for different features to be measured for each state: the basic idea is that no single representation may be valid for an entire activity and the "right" features to measure may be different at different phases of a gesture [Wilson and Bobick, 1995]. HMMs are also employed in our work on recognizing American Sign Language [Starner and Pentland, 1995]. Although this may not be considered natural gesture, it is a grammar controlled activity, much like the assembly of a device or the unloading of particular type of object: Part A must be raised before Part B can be extracted.

Our most recent work in gesture has once again moved away from HMMs and back to explicit (or visible) states. The main idea is that for gesture it is often the case that the temporal characteristics of the semantically significant gestures' are known and that one would like to devise a parsing mechanism capable of segmenting such gestures from incoming video. We have demonstrated an approach which allows us to parse natural gestures generated by someone telling a story. The system is able to identify important or meaningful gesture based upon the temporal structure [Wilson, et al., 1996] and permits the automatic selection of significant video subsequences in tele-communication situations.

4.2 Coupled activities

When employing HMMs, the system being modeled must be considered as a single process whose useful history can be summed up in the value of a single discrete variable. Many interesting systems, particularly those associated with human activity, are composed of multiple interacting processes. We have recently developed a method for coupling HMMs to model these interactions, and demonstrated their superiority to conventional HMMs in a vision task classifying two-handed actions [Oliver, et. al., 1997].

4.3 Temporal behaviors: driving

Automobile drivers' intended action (e.g., to turn, change lanes, brake, etc.) can be inferred by observing their control inputs (steering and acceleration) as they prepare to execute the action. Actions are modeled as a sequence of internal mental states, each with a characteristic pattern of driver control behavior; this is similar to the hidden Markov modeling discussed above. By observing the temporal pattern of the drivers' control inputs and comparing to the action models, we can determine which action the drivers are beginning to execute. In the case of driving the actions are events like turning left, stopping, or changing lanes. The internal states are the individual steps that make up the action, and the observed behaviors will be changes in steering angle and acceleration/braking of the car.

Even apparently simple driving actions can be broken down into a long chain of simpler sub-actions. A lane change, for instance, may consist of the following steps (1) a preparatory centering the car in the current lane, (2) looking around to make sure the adjacent lane is clear, (3) steering to initiate the lane change, (4) the change itself, (5) steering to terminate the lane change, and (6) a final recentering of the car in the new lane. We have begun to statistically characterize the sequence of steps within each action, and use the first few preparatory steps to identify which action is being initiated. Initial pilot studies [Boer, et. al., 1996] indicate that driver's patterns are quite predictable and it is possible to both know almost instantly when a drive is going to turn in a particular direction, and to know if the control is following normal statistical patterns.

4.4 Applications of activity understanding

Applications of activity understanding technologies include:

- **Advanced command and control interfaces.** One method to increase the efficiency of personnel in command and control settings is to increase the bandwidth of communication between man and machine. Gesture is well suited to noisy or cluttered environments, and can model additional attributes such as where an operator's attention is focused and his input intended.

- **Surveillance.** Coupled with movement recognition, activity detection can be used to spot suspect sequences of behavior, such as a van operator that drives a van to the front of a building but then walks away from the van.

- **Predictive behavior warning and anticipation.** Operator failure incidents can be reduced if potential accidents can be detected

before their occurrence, such as reducing convoy collisions. Alternatively, behavior prediction should be able to predict the course of evasive maneuvers quickly.

- **Behavior anomalies.** Identify vehicles moving in an unusual manner, e.g., drivers who are lost (don't know where they are going), under the influence of drugs, or vehicles that are unusually loaded.

5 Action recognition and the use of context

5.1 Approximate models

Perhaps the most difficult aspect of understanding action from video is that the action defines the visual context. For example, if someone is manipulating an object, then the best method to find his hands in the imagery might be quite different than the appropriate technique for a conversational situation. The difficulty, of course, is that the action defines the context, but that the context established the best way to see during the action.

Our recent work on approximate world modeling is designed to address this problem [Bobick and Pinhanez, 1997]. The basic idea is to use some potentially inaccurate but widely applicable general purpose vision routines to try to establish an approximate model of the world. This model, in turn is then used to establish the context and select the best vision routine to perform a given task. A fundamental innovation of this work is that approximate models can be augmented by extra-visual, contextual information. For example, a linguistic description might be available indicating an approximate position for an object. Because we assume a potentially inaccurate world model, that information can be incorporated directly.

Our initial system employed a simple inference mechanism that draws implications about visual features that might be present during a given action in a given context. Our current focus is to extend this work by deriving a "Past-Now-Future" constraint satisfaction paradigm (derived from Allen's temporal interval algebra) that would allow the system to reason about sequences of events that constitute an action [Pinhanez and Bobick, 1997].

5.2 Context-sensitive tracking

Finally, we have also been developing context-sensitive tracking methods [Intille and Bobick, 1995]. These methods are particularly useful for domains in which the relevant features for tracking vary depending upon locale or situation. The use of spatially and temporally local context greatly improves the robustness of object tracking procedures. For example, tracking a truck might be best accomplished using a heat signature when in a forest, but a shape desorption when n the open highway.

5.3 Applications of context-sensitive action

Applications of context-sensitive action recognition technique include:

- **Site monitoring.** For example, the activity of a fleet (say re-fueling) is determined by a particular temporal ordering of events, not the precise duration appearance of most of the components.
- **Battlefield surveillance.** Tracking possibly non-cooperative entities requires a tracking mechanism that can exploit knowledge about the items being tracked and the background or distractors which are not.

6 Conclusion

The project reported here is a Focused Research Effort and, accordingly, most of work performed comprises fundamental research. However, the robustness of some of the technologies developed — such as limited environment action recognition or the driver behavior interpretation — can be easily improved by limiting the tasks. The application scenarios need to be carefully specified and the demand requirements made realistic with respect to the stage of development of the technologies for them to be integrated into operational systems. However, many such restricted domains exist (e.g. monitoring a port or depot) making the action recognition technologies presented viable either today or in the near future.

7 References

Azarbayejani, A., and A. Pentland [1996], "Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features," *ICPR'96,* Vienna, Austria.

Azarbayejani, A., and Wren, C., and Pentland, A., (1996) "Real-time 3-D tracking of the human body," IMAGE'COM 96, Bordeaux, France, May 1996

Beymer, D. and T. Poggio, "Image Representations for Visual Learning." Science, Vol. , pp. 9, 1996.

Bobick, A. [1996], "Computers Seeing Action," *BMVC* Edinburgh.

Bobick, A. [1995], "Video Annotation: Computers watching video," *ACCV* Singapore.

Bobick, A. and J. Davis [1996a], "An appearance-based representation of action" *ICPR '96*, Vienna, Austria.

Bobick, A. and J. Davis [1996b], "Real-time Recognition of Action using Temporal Templates, *WACV*, Sarasota, Florida.

Bobick, A. and C. Pinhanez [1997], "Controlling view-based algorithms using approximate world models and action information," *IEEE CVPR*, San Juan, Puerto Rico.

Bobick, A. and A. Wilson [1995], "A state-based technique for the summarization and recognition of esture," *ICCV*,pp. 382–388, Cambridge, MA.

Boer, E., M. Fernandez, A. Pentland, and A. Liu [1996], "Method for Evaluating Human and Simulated Drivers in Real Traffic Situations," *IEEE Vehicular Technology Conference,* Atlanta, GA.

Brand, M., N. Oliver, and A. Pentland [1997], "Coupled hidden Markov models for copmlex action recognition," to appear *CVPR*, San Juan, Puerto Rico.

Brunelli, R. and T. Poggio, "Face Recognition: Features versus Templates." IEEE PAMI, 15, 1042-1052, October 1993.

Campbell, L., D. Becker, A. Azarbayejani, A. Bobick, and A. Pentland [1996], "Features for gesture recognition using real-time 3d blob tracking," *Int'l Face and Gesture Workshop,* Killington, VT.

Campbell, L. and A. Bobick [1995], "Recognition of human body motion using phase-space constraints," *ICCV,* pp. 624–630, Cambridge, MA.

Darrell, T. and A. Pentland [1993], "Space-Time Gestures," *IEEE CVPR*, NY, NY.

Darrell, T. and A. Pentland [1995], "Cooperative Robust Estimation with Layers of Support," *IEEE Tran. Pattern Analysis and Machine Vision*, Vol. 17, No. 5, pp. 474-487

Intille, S. and A. Bobick [1995], "Closed-world tracking," *ICCV,* pp. 672–678, Cambridge, MA.

Jones, M. and T. Poggio [1995], "Model-based matching of line drawings by linear combination of prototypes", *ICCV*, Cambridge, MA.

N.K. Logothetis, J. Pauls, H. Bülthoff and T. Poggio, "View-dependent Object Recognition by Monkeys." Current Biology. Vol. 4 No. 5, 401-414, 1994. (May/June)

Oren, M., C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio [1997], "Trainable System for People Detection," *DARPA Image Understanding Workshop*, New Orleans, LA.

Pentland, A. (1996) Smart Rooms, *Scientific American*, Vol. 274, No. 4, pp. 68-76, April 1996.

Pentland, A. and A. Liu [1995], "Toward Augmented Control Systems," *Proc. Intelligent Vehicles '95*, Detroit, MI.

Pinhanez, C. and A. Bobick [1996], "Approximate world models: Incorporating qualitative and linguistic information into vision systems," *AAAI*, Portland OR.

Pinhanez, C. and A. Bobick [1997], "PNF-propagation and the detection of actions described by temporal intervals," *DARPA Image Understanding Workshop*, New Orleans, LA.

Poggio, T. and S. Edelman (1990). "A network that learns to recognize three-dimensional objects." Nature 343: 263-6.

Starner, T. and Pentland, A. [1995], "Real-time reading of American Sign Language," *Int'l Symposium on Computer Vision*, Miami, FL.

Wilson, A. and A. Bobick [1995], "Learning visual behavior for gesture analysis," *Int'l Symposium on Computer Vision*, Miami, FL.

Wilson, A., A. Bobick, and J. Cassell [1996], "Recovering the temporal structure of natural gesture," *Int'l Face and Gesture Workshop,* Killington, VT.

Wren, C., A. Azarbayejani, T. Darrell, and A. Pentland [1995], "Pfinder: Real-time tracking of the human body," *SPIE Proc.*, Vol 2615.

Surveillance and Monitoring Using Video Images from a UAV

Gérard Medioni and Ram Nevatia
Institute for Robotics and Intelligent Systems
University of Southern California
Powell Hall Room 204, MC-0273
Los Angeles, California 90089-0273
http://iris.usc.edu/Outlines/vsam-project.html

Abstract

We present a methodology to perform the analysis of a video stream, as generated by an Unmanned Air Vehicle observing a theater of operation. The goal of this analysis is to provide an alert mechanism to a human operator. We propose to first detect independently moving objects, then to track and classify them, and to infer behaviors using context. We propose to use, as context, the information contained in a site model, which we will register with the image. We present a technical approach, together with a demonstration plan and an evaluation procedure.

1 Introduction

Continuous surveillance and monitoring in battlefield and urban environments is becoming feasible due to the easy availability and lowered costs of video sensors. Such sensors may be deployed on stationary platforms, be mounted on mobile ground vehicle, or be airborne on board Unmanned Air Vehicles (UAVs). While the multiplicity of such sensors would permit close surveillance and monitoring, it is difficult to do so by relying on purely manual, human resources. Not only would the cost of humans observing sequences from these multiple sensors be prohibitive, but unaided humans may have difficulty remaining focused on the tasks. Typically, long periods may pass before any event of interest takes place; it is easy for human attention to wander in such a situation, and significant events may be missed. Even partial automation of the process to indicate possibly significant events to a human will considerably improve the efficiency of the process. Note that the automatic analysis need not completely define the threat, but enough evaluation must be done so that false alarms can be kept within acceptable limits.

* This research is supported in part by the Advanced Research Projects Agency of the Department of Defense and is monitored by U. S. Army.

The key task of video surveillance and monitoring (VSAM) is to observe moving vehicles and humans, and to infer whether their actions pose a threat that should be signalled to the human monitor. This is a complex task and, in general, requires integration of information from multiple sensors. Further, the deployment of, and control of the sensors, may depend on the perceived events. We plan to focus on data from a single UAV (though information from multiple UAVs could be integrated).

The UAV video introduces several constraints. The UAVs fly at fairly high altitudes, so the resolution and the field of view are limited. This limits the kinds of judgements that can be made, nonetheless, we believe that several significant events of interest can be detected and used to cue a human monitor, vastly reducing the amount of data that the human needs to observe.

The most important information that can be extracted from a video sequence is that of moving vehicles in the scene. We should be able to detect these moving vehicles, estimate their speeds and trajectories, and observe their behavior (within the constraints of available resolution and time). Motion detection is made difficult as both the observer and some elements of the scene may be moving. It may be hard to estimate 3-D trajectories due to lack of resolution.

Motion by itself, however, is not a sufficient indication of a threatening or otherwise interesting activity. In most natural scenes, there is significant amount of normal vehicle motion. It is *unusual* motion patterns that are of interest. We believe that the use of a site model, and context, can help us separate the mundane from the unusual. For example, normal traffic flow on a highway should not be signalled, but abnormal speeds or a certain aggregation of vehicles may represent a significant event. Even more complex behaviors may consist of a number of vehicles or humans acting cooperatively, and of the pattern of these activities. The actual behaviours of interest will be decided upon by consultation with the user community.

Our approach requires us to have at least crude models of the site being observed; for the monitoring system to also have an ability to recognize features such as roads and buildings generically is beyond the scope of this effort. In addition, we will need models of what is normal and unusual behavior, and how it depends on context.

Our approach consists of three major steps. The first is the detection and tracking of moving objects, the second serves to relate these vehicles to features known in a map or site model, and the last is used to infer the behaviors.

As we must deal with moving objects and moving observer, we plan to first detect egomotion, which is manifested globally. Egomotion is then used to register frames to detect independently moving objects, and to track them. Image to model correspondence will help in relating the observed motion to relevant features on the ground. We expect techniques using low level features such as lines and curves to suffice for matching in this task. As video images come continuously, tracking from frame to frame is a much easier task than looking at isolated frames. Finally, behavior analysis will be based on the interaction between vehicle trajectories and the features around them. Certain speeds and trajectories in certain contexts indicate a threatening behavior. More complex behaviors will be analyzed by observing actions of groups of vehicles, rather than just single vehicles.

Each of these steps poses significant image understanding (IU) challenges. While there is much knowledge in the field, that is relevant to solving such problems, the techniques have not been put together to yield complete systems.

As a means of background, we start by describing the elements of such a complete system, the EPSIS Billboard Replacement System, which shares some common characteristics with our scenario, then proceed with the details of our technical approach.

2 The EPSIS Billboard Replacing System

2.1 Background

It is a common practice to place billboards advertising various products and services during sports events. These billboards target not only the spectators at the stadium, but also (and mostly) the viewers of the TV broadcast of the event. This fixed advertising is therefore limited, as the billboards might be advertising products out of context for the TV audience, especially for international events.

We are presenting a system to automatically substitute, in real-time, one billboard by another, synthetically created, billboard. It aims at replacing the billboards in the scene in such a way that it should be transparent to the viewer. It allows a local TV station to plant its own advertisement billboards regardless of the original billboard, thus increasing the overall effectiveness of the advertising.

The process by which we accomplish this goal is therefore the composition of a video stream and a still image, to create a new, smoothly blended and photo-realistic video stream.

Editing of images or image streams is fast becoming a normal part of the production process[1]. Many recent movies, such as Terminator 2, Forrest Gump, Casper, ID4 seamlessly blend live images with Computer Generated Imagery. The mixing of multiple elements is performed primarily by *screen matting,* in which the background is of almost constant color, generally blue or green. This approach requires a very controlled studio environment and operator intervention for optimal results.

Our system must instead function without active cooperation, in real-time (therefore automatically, without operator intervention), and in a non controlled environment. Furthermore, it must also adapt the model to fit the observed billboard. It involves the "intelligent," automatic manipulation of images and image streams, *based on their contents.*

The system receives as input a TV broadcast signal, must identify a given billboard in the image flow, track it precisely, and replace it with another pattern (fixed or animated), broadcasting the replaced signal, in real-time, with only a short, constant delay. Figure 1 presents an example frame of billboard

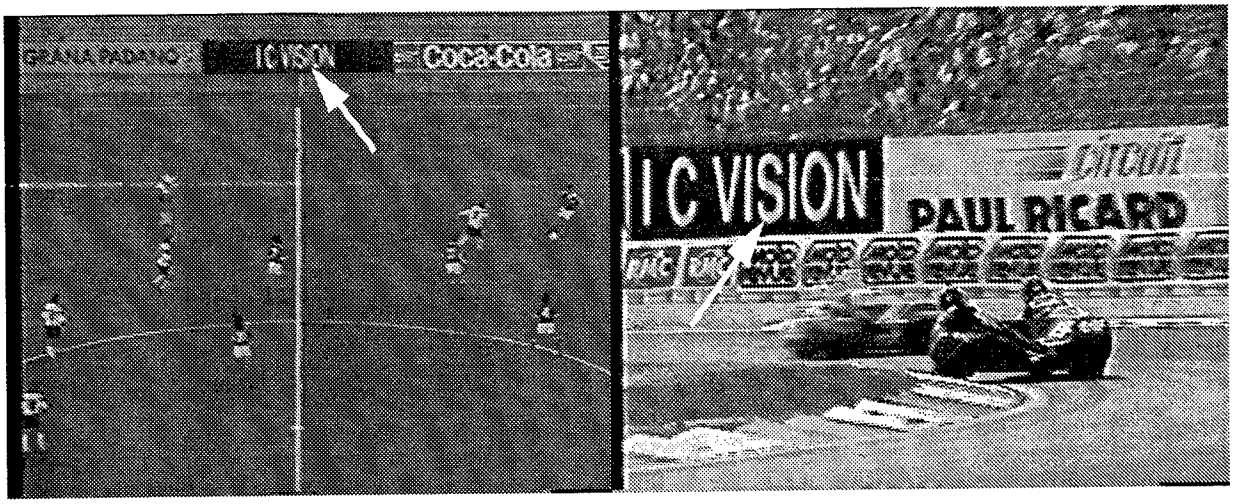


Figure 1Two examples of billboard replacement in a Video Sequence

replacement.

2.2 Requirements and challenges

The fundamental requirement that the system perform *on-line* in *real-time*, imposes major constraints on the design and implementation:

- No human intervention is possible.
- No on-screen errors are permitted. The system has to include self quality control mechanisms to detect problems and revert to the original signal when they occur.
- Complex high level algorithms are limited due to the need for implementation in real time.
- No cooperation from the field is expected, in order to allow the system to operate independently from the imaging process (e.g. at the down link).

The contribution of such a system, for which a patent was issued[2], thus resides both in the design and implementation of the individual modules (finder, tracker, replacer), and in the management of failure and uncertainty for each of these modules, at the system level, resulting in reliable replacement.

2.3 Implemented Solution

2.3.1 Overall System Design

The task of the system is to locate a planar, rectangular target billboard in the scene, detect camera switches, track the billboard throughout the sequence (between camera switches), and replace it with a new billboard. The direct naive approach would be to inspect the incoming frames, search for the billboard and replace it. Unfortunately, this approach is not sufficient, as it may be impossible to locate the billboard in the current frame: This may be due to large focus or motion blur, or to the billboard being occluded, or to the fact that only a small part of it may be in the field of view. The billboard may therefore be found only in a later frame of the sequence, and it is not advisable to start replacing then, as this would be offensive to the viewer. Instead, replacement should be performed on the whole sequence to avoid billboard switches on screen.

Our system relies on modular design, and on a pipeline architecture, in which the search and track modules propagate their symbolic, low-bandwidth results throughout the pipe, and the replacement is performed at the exit of the pipe only, therefore relying on accumulated information. This allows the system to make replacement decisions based on complete sequences, thus avoiding mid-sequence on-screen billboard changes.

The ***Finder*** module searches for the target billboard in the entering frames and passes its results to the Updater, which propagates them throughout the buffer. It first extracts "interesting" points (corners, or other "busy" formations) in the image, then selects the interest points which are most likely to come from the target billboard based on color information. It then finds a set of corresponding points between model points and image points, using an affine-invariant matching technique proposed by Lamdan and Wolfson[5], and uses these correspon-

dences to find the precise (to a sub-pixel resolution) location of the billboard.

The ***Global Motion Tracker (GMT)*** module estimates the motion between the previous and current frames, *regardless of whether the billboard was found or not.* This is used as a mechanism for predicting the billboard location in the frames in which it was not found. The prediction is necessary to ensure continuity of replacement, since we do not want the billboards to switch back and forth between the original and the new one in front of the viewer. The GMT also performs the task of camera switch detector.

Since we are interested in the motion of the camera and not in a per pixel motion, we take a global approach, and use an iterative least squares technique on all pixels of the image[3]. The images are first smoothed and the spatial and temporal derivatives computed. Using this information, estimates of the motion parameters are computed. Using these estimates, Frame $t+1$ is warped towards Frame t, and the process is repeated. Since Frame $t+1$ gets closer to Frame t at every iteration, the motion parameters should converge. The accumulated parameters are then reported to the Updater. We have implemented the algorithm at multiple levels of resolution. A Gaussian pyramid is created from each frame[4]. At the beginning of a sequence, the algorithm is applied to the lowest resolution level. The results from this level are propagated as initial estimates for the next level up, up to the highest level. This allows for recovery of large motions.

An improvement to the global motion algorithm allows for accurate and stable results, even in the presence of independently moving obstacles in the scene. This is achieved by scaling the coefficients of the motion equations inversely proportional to the temporal derivatives. Moving obstacles do not match when the images are warped according to the camera motion. Therefore, pixels corresponding to obstacles produce high temporal derivatives, and consequently contribute less. The improved results allow for long propagation of estimates along the sequence.

The ***Replacer*** performs the graphic insertion of the new billboard, taking into account variations from the model due to lighting, blur and motion.

Given the coordinates of the billboard corners in the current image, the Replacer module replaces the image contents within these corners (the billboard) with the new desired contents (usually a new billboard). Because the human eye is quite sensitive to sharp changes in colors, we correct the gain and offset of the replaced billboard to make it appear close to the average intensity of the image. Note that we currently assume that the original billboard is unoccluded. Mechanisms which allow for detection of obstacles in front of the billboard are currently under development with promising results.

The ***Updater*** handles communication within the buffer and also manages the *Measure Of Belief (MOB)* associated with the information passed along, due to the MOB of each of the modules, and a decay related to the length of the propagation. The information about scene changes is also used so that the *Updater* does not propagate the predictions beyond the scene change markers.

It collects data from all the other modules, and corrects missing or inaccurate information within a processed sequence. We can visually think of the system as a circular buffer, holding a frame and a frame attribute in each of its cell. The Updater manipulates these attribute records only, which are composed of a small number of parameters, and processes *all* attribute records in the buffer in one frame time.

Figure 2 presents the overall system architecture. As the frame at time t comes in from the video source on the right, the Finder searches for the billboard. At the same time, the Global Motion Tracker (GMT) computes the camera motion between the previous and current frames, and stores it in an attribute record. If the billboard is found, its four corners are recorded in the attributes record, and the Updater unit predicts the location of the billboard in all the (previous) frames from the first frame of the sequence to frame $t-1$, based on the computed motion, and updates the attribute records accordingly. As the frame is about to be displayed, the Replacer performs the insertion.

Let us consider the difficult case where the billboard is slowly entering into view, as a result of a pan or zoom. In this case, the billboard cannot be found initially by the Finder. As the frames continue to come in, the Global Motion Tracker computes the camera motion between frames, regardless of whether the billboard was found or not. The camera motion parameters found are stored in the frame attribute record to be accessed by the Updater. When the billboard is reliably found in

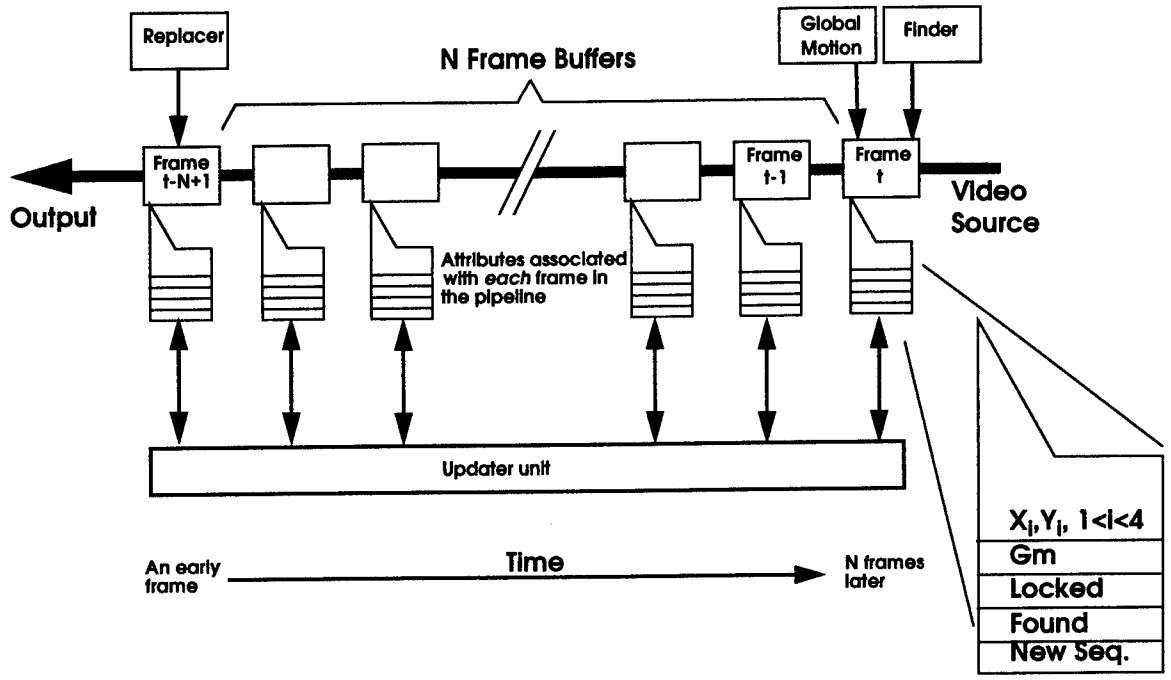


Figure 2A block diagram of the system.

some frame, *t*, of the sequence, the Updater module uses the motion parameters computed earlier, to predict the location of the billboard in all the frames from the first frame of the current sequence up to frame *t-1*. Since this is a very simple computation (not image based), involving low bandwidth communication, it can be performed for the whole buffer in one frame time. As the images reach the end of the buffer, we know the location of the billboard, either directly form the Finder, if it was found in this frame initially, or via a prediction from the Updater, using the motion information.

The combined use of the Global Motion Tracker, the delay buffer and the Updater mechanism, allow the system to, in essence, go back in time without having to process the image again, and to use information from the current frame to locate the billboard in earlier frames. This enables the system to perform well under varied conditions, such as occlusion and entering billboards. The system is also very robust to failure of specific modules, as it can overcome failure in some frames by using information from the other frames of the sequence. It is important to note that each image is processed once only, and that each module works at frame rate, thus the system works in real-time, introducing only a constant delay.

This design can guarantee that no offensive substitution will take place, as long as a whole sequence fits in the buffer. Otherwise, in case of a problem occurring after replacement is started, a smooth fade back to the original billboard is used. In practice, a buffer of the order of 3 seconds (180 fields in NTSC), covers a large percentage of sequences in which the billboard is present.

2.3.2 The Machine

A design somewhat simpler than the one described here has been made operational by Matra CAP Systèmes using off-the-shelf components, and used by Symah Vision for live broadcasts.

This successful aggregation of computer vision and computer graphics techniques should open up a wide avenue for other applications, which are either performed manually currently, or simply abandoned as too difficult.

On a different note, it is interesting to note that such a system also casts some doubts as to the authenticity of video documents, as predicted in fiction such as Rising Sun. It shows that digital video documents can be edited, just like audio and photo documents.

3 Research Issues and Approach

We plan to develop a system for analysis of video image sequences from a single Unmanned Air Vehicle (UAV) with the objective of detecting impor-

tant events that present a threat or are significant in other ways and alert a human monitor to them. Our goal is not complete automation but reliable operation while minimizing false alarms for the human, resulting in a great reduction on the time that the human must devote to monitoring such video streams.

The most relevant information that can be extracted from a video sequence is that of moving objects in the scene. We therefore propose to process the video stream to:

- estimate image motion due to the observer (egomotion), and compensate for it,
- detect regions in the image whose motion differ from the above,
- track these tokens over time,
- infer behaviors from this analysis.

The overall approach to the problem is depicted in schematic form on Figure 3. The modules correspond to the major tasks mentioned above.

Note that these tasks require different time frames: while it is possible to estimate egomotion from only 2 frames, reliable tracking of independent objects requires several frames, and behavior inference demands an even longer aggregation of frames.

We now describe our technical approach in some detail..

3.1 Motion estimation and segmentation

We need to detect motion that is independent from the flow induced by the sensor (egomotion) in the image stream. To accomplish this, we propose to first estimate the egomotion, use it to register frames and detect independently moving objects and then to track them to compute their trajectories. We describe these steps below.

3.1.1 Egomotion estimation

Since we are interested in the motion induced by the camera, and not in a per pixel motion, we take a global approach, and use an iterative least squares technique on all pixels of the image [1, 7, 14]. The method, therefore performs the task of *image stabilization*, assuming the sensor motion is limited to pan, zoom and tilt.

However, the motion model may not be able to take reflect the variations of displacements of the object features due to its 3-D geometry. In our scenario,

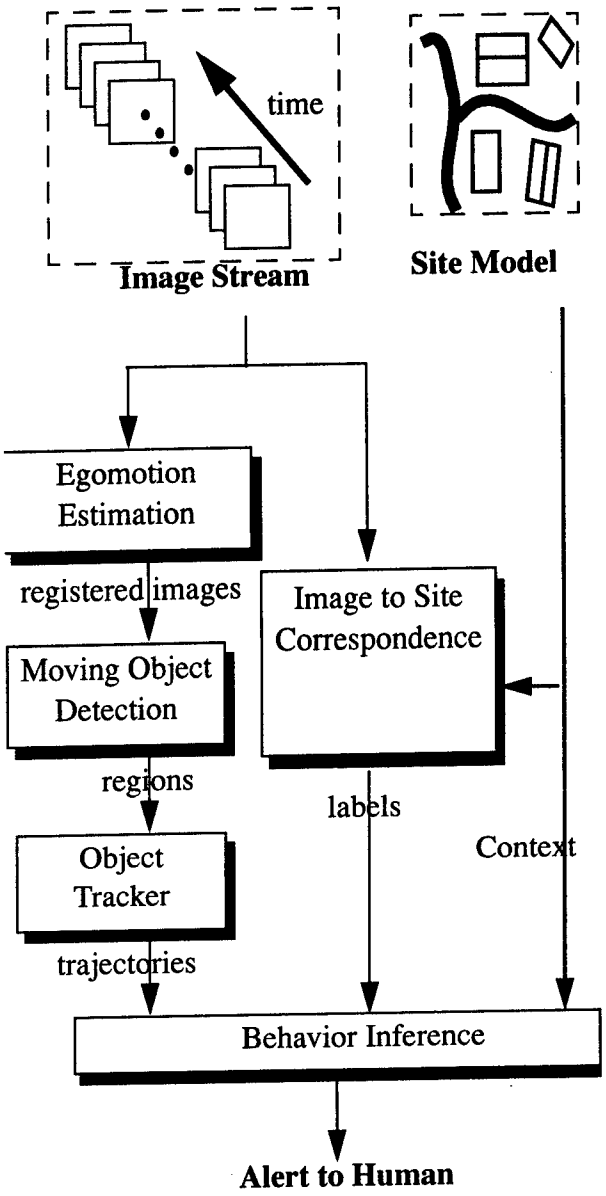


Figure 3Overview of the Complete System

the sensor is relatively far from the object. Therefore the deviation of the point-of-view, which produces these displacements, is small. Furthermore, even these small deviations can be corrected with coarse knowledge of the terrain, as obtained from the site model, if necessary.

We have also adapted the basic algorithm to function in the presence of independently moving objects in the scene. This is obtained by scaling the coefficients of the motion equations inversely proportional to the temporal derivatives. Moving objects are not registered when the images are warped ac-

cording to the camera motion. Therefore, the pixels corresponding to objects have high temporal derivatives, and consequently less weight in the coefficients. A by-product of the algorithm is the identification of independently moving pixels.

3.1.2 Tracking

Pixels identified by the egomotion module as not coherent with the induced displacement are marked as belonging to mobile objects. Given the resolution we can expect from the sensor, we anticipate these mobile objects to consist of only a few pixels in the image. It is therefore unreasonable to expect to try and perform 3-D structure estimation for these objects[16]. We propose instead to represent them only as 2-D regions, and track these regions in time [8, 15]. For vehicles moving on the ground, the image displacement between two frames will be small, but temporally coherent. We will therefore perform a multiframe analysis of the motion, for robustness and accuracy, using a 2-D translational motion model for the image features.

When airborne vehicles, such as helicopters and fixed wing airplanes are observed, the image induced displacement is much larger. The regions corresponding to these objects will also be larger, since they are closer to the sensor. In such a case, we will use the structure of the region contour to disambiguate the tracking process.

3.2 Image to Site Correspondence

To interpret the motion of the observed vehicles, it is useful to geolocate them in reference to the known features of the site. It is important to know if the vehicles are on a road or in the vicinity of certain important buildings. We consider the problem of actually detecting features such as roads from the image sequence itself without prior models to be beyond the scope of this effort. Instead, we propose to make correspondences with prior maps or site models. Approximate locations may be determined just from the knowledge of the observer vehicle parameters. We can expect to obtain highly accurate estimates of the observer location from the GPS navigation system; orientation parameters may have somewhat less precision. However, we cannot expect the navigational parameters to be accurate enough to predict exactly where a feature of interest, such as a road, might be, but we believe that simple image to map (model) matching techniques [6, 11] can suffice to bring them into accurate correspondence.

We will need to continually update the correspondences between the observed features and the map/model features. Since the data is available to us in a continuous stream, the updating process can be much simpler than one of initial correspondence. At each step, we can predict the amount of displacement and can correct by using only a small number of features.

3-D site models, even if they are not very accurate or complete, would help in the process of correspondence. However, it may be possible to make do with 2-D maps if the terrain is relatively flat and the flight of the vehicle is level.

3.3 Behavior Inference

After various vehicle and human motions have been detected and tracked, and some correspondence established between the images and the site features, we still need to interpret the motion to decide if a significant event has taken place. A first step in this process is that of motion interpretation itself. We should be able to tell whether the moving object is on the ground or is airborne as ground objects have some constraints on their motion. We can also estimate the vehicle speed which may provide some constraints on the class it belongs to (*e.g.* tanks don't travel at 100 km/h).

The next step is to try to determine if a significant event is taking place. We will study the kinds of events human monitors detect and the cues they use to detect them. Some examples are: abnormal speeds or trajectories, activity in forbidden areas, and certain kinds of group activities.

We believe that these kinds of activities can be detected by representing the expected behaviors in a symbolic template representation and verifying if the template criteria are satisfied.

4 Evaluation Plan

We describe some proposed metrics and an evaluation methodology below.

4.1 Metrics

We propose the following metrics:

1) **Detection rate:** What is the percentage of correctly recognized events? Obviously, only events that are visible, have sufficient resolution and duration can be detected.
2) **False Alarm rate:** This measures the frequency of mistaken detection.

4.2 Testing

We intend to develop and test our algorithms directly on data from operational UAVs. We expect that such data will become available to the VSAM research community. To make evaluations, we will need some *ground-truth* to compare with. For UAV data, we may not be able to get the actual ground truth, instead, we may need to rely on the judgement of human observers to see what events they are able to perceive and what their interpretation of the behaviors is.

4.3 Demonstration Plan

Our proposed research is to analyze image sequences observed from a UAV. As we are not likely to have access to UAVs in the field, we will need to demonstrate on stored images in our laboratory. We expect that sample imagery will be available from on-going UAV projects and from the IFD platform. For the early parts of our development, we should be able to use video images available from helicopter flights and other sources that are commonly used in motion analysis experiments.

Given suitable image (and other data), we expect to show the capabilities of detecting vehicles, tracking them through the sequence and indicating when the system believes the behavior to be abnormal. The output of the system can be displayed graphically and also tabulated for comparison with human assessments.

Our processing may not be necessarily at real-time speeds though computational efficiency will be a major concern. In later phases of the project, we can use the IFD platforms, if available, for real-time demonstrations.

References

[1] R. Fielding, *The Technique of Special Effects Cinematography*, Focal/Hastings House, London, 3rd edition, 1972, pp. 220-243

[2] G. Medioni, G. Guy, and H. Rom, *Video processing system for modifying a zone in successive images*, U.S. Patent # 5,436,672, July 1995.

[3] J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg, *A Three-Frame Algorithm for Estimating Two-Component Image Motion*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 9, pp. 886-896, Sep. 1992.

[4] P.J. Burt, *Fast Filter Transforms for Image Processing*, Computer Graphics and Image Processing, Vol. 16, pp. 20-51, 1981.

[5] Y. Lamdan, J. Schwartz, and H. Wolfson, *Affine Invariant Model-Based Object Recognition*, Robotics and Automation(6), 1990, pp. 578-589.

[6] P Anandan, P. Burt, K. Dana, M. Hansen, and G. van der Wal, "Real-time Scene Stabilization and Mosaic Construction," *Proceedings of the ARPA Image Understanding Workshop*, 1994, pp. 457-465.

[7] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg, "A Three-Frame Algorithm for Estimating Two-Component Image Motion," *Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, Sep 1992, pp. 886-896.

[8] W. Franzen, "Structure and Motion from Uniform 3D Acceleration," in *Proceedings of the Workshop on Visual Motion*, IEEE Computer Society, 1991, pp. 14-20.

[9] W. Franzen, "Structure from Chronogeneous Motion: A Summary," in *Proceedings of the DARPA Image Understanding Workshop*, San Diego, CA, January 1992.

[10] S. L. Gazit and G. G. Medioni, "Multi-Scale Contour Matching in a Motion Sequence," *Proceedings of the DARPA Image Understanding Workshop*, 1989, pp. 934-943.

[11] A. Huertas, M. Bejanin and R. Nevatia, "Model Registration and Validation," in *Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images*, April 1995, Ascona, Switzerland, pp. 33-42.

[12] M. Irani and P. Anandan, "A Unified Approach to Moving Object Detection in 2D and 3D Scenes," *Proceedings of the DARPA Image Understanding Workshop*, 1996, pp. 707-718.

[13] Y. C. Kim and K. Price, "Improved Correspondence in Multiple Frame Motion Analysis," in *Proceedings of the DARPA Image Understanding Workshop*, San Diego, CA, January 1992.

[14] R. MacGregor, T. Russ and K. Price, "Knowledge Representation for Computer Vision: The VEIL Project," *Proceedings of the ARPA Image Understanding Workshop*, 1994, pp. 919-927.

[15] G. Medioni, G. Guy, and H. Rom, U.S. Patent #5,436,672 *Video Processing System for Modifying a Zone in Successive Images*," awarded July 1995

[16] G. Medioni and R. Nevatia, "Matching Images Using Linear Features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):675–685, November 1984.

[17] N. Milhaud and G. Medioni, "Learning, Recognition and Navigation from a Sequence of Infrared Images," *Proc. of International Conference on Pattern Recognition*, Jerusalem, Israel, October 1994, pp. 822-825.

[18] T. A. Russ, R. M. MacGregor, B. Salemi, K. Price, and R. Nevatia, "VEIL: Combining Semantic Knowledge with Image Understanding," *Proceedings of the DARPA Image Understanding Workshop*, 1996, pp. 373-380.

[19] H. Sawhney, S. Ayer, M. Gorkani, "Model-Based 2D and 3D Dominant Motion Estimation for Mosaicing and Video Representation," in *Proceedings of International Conference on Computer Vision*, June 1995, pp. 583-590.

[20] H. Shariat and K. Price. Motion estimation with more than two frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):417–434, May 1990.

[21] C. Tomasi and T. Kanade, "The Factorization Method for the Recovery of Shape and Motion from Image Streams," in *Proceedings of the DARPA Image Understanding Workshop*, San-Diego, California, January 1992, pp. 459-472.

Extra Sets of Eyes

Robert C. Bolles, Kurt G. Konolige, Martin A. Fischler
Artificial Intelligence Center, SRI International
333 Ravenswood Ave., Menlo Park, CA 94025
Telephone: 415-859-4620, 415-859-2788, 415-859-5106
E-MAIL: bolles@ai.sri.com,konolige@ai.sri.com,fischler@ai.sri.com,
HOMEPAGE: http://www.ai.sri.com/~eyes

Abstract

The objective of this Video Surveillance And Monitoring (VSAM) Focused Research Effort (FRE) project is to develop a set of portable (the size of a baseball), inexpensive (less than $500 in production), self-contained (sensors, processing, radio, and battery) Small Sentry Modules (SSMs) that can autonomously detect, characterize, classify, and report moving objects, such as people, animals, and vehicles. These sensors will be small enough so that a warfighter could carry several of them and deploy them as extra sets of eyes, instead of using people, to monitor key areas for possible approaching threats. When a module detects a significant event, such as a person running along a path, it radios a description of the event to the warfighter and/or combat unit, providing them with advanced warning of possible danger. Numerous modules could be used to establish a sentry network in which the sensors cooperate to verify detections and estimate the 3D locations and velocities of moving objects.

1 Introduction

The nature of war has changed in several ways. First, many of the 'conflicts' are "missions other than war" or Military Operations in Urban Terrain (MOUT), as opposed to conventional warfare. Second, the number of warfighters in the U.S. armed forces has been reduced significantly, increasing the tactical importance of each one. And third, the death of a warfighter, although never 'acceptable,' is less tolerable today than in the past. Two consequences of these changes are an increased need to protect each warfighter and a renewed interest in force multiplication. In this project, we propose to develop a sequence of increasingly competent SSMs that can help fulfill both needs – protect the warfighter and enhance his effectiveness.

Our vision is to develop SSMs that are sufficiently competent to be used by a warfigther as an extra pair of eyes to watch his backside, while he continues with his primary mission. If a warfighter is clearing a building in a MOUT scenario, she can leave one device at the entrance to warn her if someone enters the building. If she's on a trail, she set up one on a tree to warn her of someone approaching from the rear. In addition, a warfighter can use one of the modules as a periscope to look around a corner and into a tunnel without exposing herself to enemy fire. For this application, the warfighter attaches her head-mounted display to the module's high-definition video output and visually explores the hidden area by physically moving the module to scan it. The ultimate modules will have sensors that combine infrared (IR) and image-intensified data to provide an excellent night-vision capability.

We expect the deployed modules to incorporate multiple sensor modalities, such as stereo, motion, radar, and acoustic, in order to maximize their applicability and minimize their errors. For example, a combination of a radar sensor to detect movement and a stereo sensor to

construct 3D descriptions of the moving object will be one of the first devices to be considered. For some applications, a combination of acoustic and visual sensors is appropriate. For other applications, inclusion of chemical and biological sensors would add a critical new dimension.

SRI has already developed the critical stereo sensor that makes this project feasible – the parts cost is approximately $200, in large quantities, and this price can be reduced further if special chips are fabricated. Within the first six months of the project, we plan to construct 20 copies of an initial stereo-based SSM, implement a development environment for it, and then distribute sets of the SSMs and the environment to the Integrated Feasibility Demonstration (IFD) contractor and the other FREs as a common experimental device.

An enhanced version of the SSM will include techniques for multiple modules to cooperate, setting up communication channels, a common coordinate system, and a procedure for operating together to verify detections and estimate their 3D locations. And our vision of the ultimate SSM is a mobile sensor package that can be used either to explore ahead of a warfighter or to trail behind, checking for potential danger signals.

2 Research Questions

Key scientific questions occur at two levels in this project. First, there are questions related to an individual SSM, such as how it detects and tracks moving objects. And second, there are questions related to networks of sensors cooperating to perform a single task, such as guarding a compound. For an individual sensor, the basic research questions focus on the following five topics:

Tasking – How does a military person specify what the SSM should "look for?" Is it "built-in?" Or are there on-site specifications required? How does the sensor know where it is relative to key functional items, such as the ground plane, roads, or hallways?

Detection – How does an SSM detect significant motion – directly (through motion analysis or doppler radar) or indirectly (through differential changes in location)? Key to a successful system is an architecture that supports cross-sensor fusion to minimize false positives and false negatives. Although many researchers have talked about strategies for such fusion, this version of the problem is unique in that the sensors must be inexpensive and operate in a complex, close-in 3D environment.

Tracking – How does an SSM continuously track and model a detected object. A key aspect of our approach is an analysis of sequences of detections for building "minimum description" models of the tracked objects.

Characterization – How does an SSM describe a tracked object in terms of its 3D size, 3D velocity, and articulation. These characterizations are critical for classification and recognition, which provide the warfighter with information at the level he can understand.

Classification – How does an SSM classify tracked a object into categories, such as person, animal, or vehicle. {Note that the SSMs do not have to distinguish civilians from warfighters to be useful.}

Communication – Once a significant event has been detected, how does the SSM communicate the information? Should send and alarm followed by a snapshot of the detected object?

For networks of cooperating SSMs, the following topics are of particular interest:

Cooperation – Can two, three, or ten SSMs working together dramatically simplify the segmentation problem in dynamic worlds? Can multiple SSMs produce significantly more complete characterizations of objects than traditional techniques? If so, at what

level do the cooperating SSMs have to communicate? Do they need to produce a single consistent model or can they each maintain partial models, share their results, and compute more comprehensive object descriptors than current single-sensor approaches? Can a set of SSMs simplify the detection and description of articulations, which are key to object classification, particularly for classes such as animals, people, and military vehicles?

Tasking – Can a set of SSMs figure out their relative positions automatically by watching people and vehicles moving in their fields of view? Or does someone have to specify their approximate locations and fields of view?

Communication – To what extent do the SSMs need to communicate among themselves. For some applications, it may be possible to directly connect them to a network. How would that change the approach to cooperation?

3 Approach

We have designed a four-staged approach for developing, testing, evaluating, and demonstrating SSMs. The stages are:

Develop and Deliver Small Stereo and Motion Sensors – In the first 6 months of the project, we plan to implement a small stereo and motion sensor, document it, and distribute 15 to 20 copies of it to other VSAM contractors for evaluation. SRI has already developed a creditcard-sized stereo system, called the Small Vision Module (SVM). We plan to add motion analysis to this device and then distribute copies of it.

Develop a Basic SSM – We will develop algorithms for detecting, tracking, characterizing, and classifying moving objects observed by an SSM.

Develop an Enhanced SSM – We will explore techniques for adding additional sensor modalities, such as sonar, acoustic, infrared, or radar to an SSM.

Explore Cooperating SSMs – We will explore techniques for using several SSMs to cooperatively perform a task, such as monitor a perimeter around a compound.

In the optional years of this project, we plan to continue research to enhance the quality, reliability, and transferability (into an IFD or other DoD programs) of our results. At the end of three years, we plan to demonstrate an SSM designed for a specific user. Our intent is develop and deliver an initial prototype that a user can experiment with and provide feedback to guide future development. At the end of four years, we will demonstrate a second version of a user-specific SSM. At the end of five years, we will demonstrate a network of cooperating SSMs.

4 Plans for Evaluation

As described above, we plan to demonstrate, test, and evaluate a sequence of increasingly competent SSMs. In the first six months of the project, we plan to distribute a set of SSMs to the IFD contractor and the FREs for evaluation. Feedback from these experiments will help shape the next version of the module, which we will demonstrate at the end of a year and a half. At the end of two years, we will demonstrate techniques for using multiple SSMs cooperatively to increase the reliability and robustness of detections and to increase the accuracy of the computed properties of detected objects.

For each version of the module, we plan to characterize its effectiveness along five dimensions:

Types of Applications – What types of tasks can the SSM perform? Under what conditions (lighting, weather, scene content, etc.) does the SSM operate effectively?

Time for Tasking – What is involved in deploying an SSM? Does the person have to specify parameters of its task or indicate regions of interest?

Range of Objects Detected and Classified – What types of objects can be detected,

tracked, characterized, and classified? For example, can the SSM identify people, vehicles, and moving shadows? Can it distinguish people walking from people running?

Detection and Classification Statistics – How frequently does the SSM miss a key object? How often does it generate false alarms?

Potential Cost – What is a predicted production cost for the SSM?

5 Summary

We believe that the SSMs will save lives by dramatically increasing the "safety zone" around a warfighter and reducing his exposure to hostile fire. The modules, when deployed as sentries, automatically detect, track, characterize, and warn the warfighter of approaching people or vehicles, while ignoring blowing leaves, shadows, and small animals. When deployed as a periscope, these modules provide night-vision capabilities to the warfighter without exposing him to potential harm. In summary, these devices can multiply the effectiveness of a force by providing the eyes (and ears and noses) for critical tasks, and thus freeing the warfighters to concentrate on their primary tasks.

A Forest of Sensors

E. Grimson P. Viola O. Faugeras T. Lozano-Pérez T. Poggio S. Teller
Massachusetts Institute of Technology Cambridge MA

Abstract

This project focuses on utilizing large sets of inexpensive sensors to monitor activities within a site. This forest of sensors will self-calibrate, build approximate site models and classify and monitor activities within the site.

This PI Report describes work that will be conducted under a newly issued grant as part of DARPA IU's VSAM project.

1 Top Level Objectives

The rate of advances in low power micro-electronics suggest that soon cameras, processors, and power supplies will be cheap, reliable and plentiful. We want to be ready for a time when it will be possible to build a complete, autonomous, vision module (AVM) using only a few chips. Such a device will be able to function autonomously for days or weeks at a time, sending information over low bandwidth radio connections. With the addition of a small solar array, an AVM might operate indefinitely.

Integrated with a steerable platform, AVM's can perform autonomous surveillance and make critical visual observations from locations which are simply too dangerous for personnel. But there is another, perhaps more speculative, niche for extremely cheap AVM's. A *disposable* AVM (dAVM) would be entirely solid-state and have no moving parts. It would be the size of a grenade, a good deal lighter, and just as tough. Dozens if not hundreds could be dropped from planes, scattered throughout a field, or mounted on the rear of every vehicle. A dAVM could act anywhere that an extra pair of eyes might be useful: protecting a perimeter; in surveillance operations; or directing fire. We envision such a collection of dAVMs as a **FOREST OF SENSORS**, and we believe that developing the capabilities to deploy and most importantly to process the data from such a forest of sensors will revolutionize existing surveillance and monitoring methods.

While we are interested in designing dAVMs, we believe that one can focus on their utilization largely independently of their design and instantiation. Thus, one can posit the existence of such dAVMs, and ask what activities are made possible by their availability. Many scenarios involving surveillance require monitoring of large amounts of imagery from many vantage points. In short, significant manpower must be expended. Imagine instead, a scenario where 100 dAVM's could be rapidly affixed to trees, rocks, or other elements in an environment. Alternatively, imagine a suite of dAVMs mounted on small drone aircraft. The dAVM's would work in concert, dividing the task of observation automatically among themselves. Together they would immediately identify gaps in the area that can be observed and suggest placement of additional devices. The forest could detect failures of individual units, and use the inherent redundancy of multiple sensors to avoid failure of the ensemble. The forest could use change detection in combination with focus of attention methods to allocate resources to components most able to use them. Given such a forest of small, cheap, robust sensors, a large number of important surveillance tasks becomes feasible, including:

Perimeter Patrol: Protecting a temporary camp's perimeter is a difficult enterprise. Patrols must be organized and a perimeter established. If instead, we have a forest of sensors that have been attached by troops to points on the perimeter, then the surveillance and patrol can be heavily automated. Equipped with low light sensors, each dAVM would detect motion and classify it (e.g. animal vs. vehicle). If animal, further analysis about location and type could be performed. When enemy incursion is detected, sentries would be immediately advised of the situation. The dAVM's would be so inexpensive that they need not be retrieved, but can continue to observe and report activity.

Visual Mines: Modern mines are unfortunately non-discriminatory. They will explode when trig-

[0]This report describes research supported in part by ARPA under ONR contract N00014-94-01-0994, and by DARPA contract TDB. PIs may be contacted at welg@ai.mit.edu, viola@ai.mit.edu, faugeras@ai.mit.edu, tp@ai.mit.edu, seth@ai.mit.edu. URL for project available at `http://www.ai.mit.edu/projects/darpa/vsam/`

gered by enemy forces, by civilians and by friendly forces. Imagine replacing explosive mines with "visual mines". dAVM's could be placed along key lines of passage, and equipped to trigger a burst of communication to remote observer sites when human activity or vehicle activity is recorded in the vicinity of the mine. This would enable a remote operator to identify friend or foe based on the visual data relayed by the visual mine, to alert nearby troops to investigate, or to coordinate with nearby fire control centers. This is not just a case of placing motion detectors, since such simple systems are likely to overwhelm the operator with tons of false positives. Visual mines should have some "intelligence" so that they only alert the operator when they detect instances of likely activity, based either on training from the operator or on generic models of activity.

Urban Security: In urban surveillance and monitoring, a forest of sensors will: i) register different viewpoints and create virtual displays of the facility or area; ii) track and classify objects (people, cars, trucks, bags, etc.); iii) overlay tracking information on a virtual display constructed from the observations of multiple cameras; iv) learn standard behaviors of objects; v) selectively store video. Low bandwidth tracking information could be continually stored allowing the observer to query the system about activities: "What did the person who left this bag do from 2 minutes before until 2 minutes after leaving it?" "Where is that person now?" "Show me a video of that person." Tracking information could be used to tag activities such as cars speeding towards the facility, people climbing perimeter walls, and unusual loitering near a facility.

In all these cases, it will be difficult to carefully place and calibrate each dAVM. A dAVM must discover its position and its relationship to other dAVM's. Executing geometric self-calibration will enable coordination of the dAVMs to build approximate site models. But the dAVMs must also cooperate in their observations of moving objects. Some version of GPS will be helpful here, but it will not solve the entire problem. In order to work in concert, dAVM's will need more than simple camera calibration, they will need a notion of *activity calibration.*

Even with coordination between dAVMs, the job of monitoring activity is not complete. Imagine dAVMs observing two different villages: in one opposing forces march through a square; in another civilians congregate. Detecting the difference is critical in modern engagements, where losses and civilian casualties must be kept to an absolute minimum. Many features differentiate these scenarios: the uniforms, the weapons, the activity patterns. The latter points to a critical missing component in surveillance, the ability not just to detect basic units of activity but to automatically interpret them. This is a difficult task which is very different from the tasks of change detection and target identification. Much of the success of target recognition has been based on the elegant use of geometric models. Unfortunately, there can be no static geometric model of a platoon walking through a wood, the construction of a revetment, or the passing of a military convoy. In fact, the fundamental question has been changed: target recognition finds objects; what we need is *activity recognition.* Such recognition should complement and enhance static analysis with dynamic analysis.

2 Technical Challenges

Two major technological advances are necessary to enable the widespread deployment of dAVM's: – we need techniques for fusing visual information observed at different times and from different locations; – we need a framework in which to construct *activity models* so activity can be reliably and efficiently detected and interpreted.

Processing the information generated by many AVM's as they observe a locale will be quite different from the processing performed on a single image. For AVM surveillance, we would like to detect some activity occurring in the real world at real-time. Not only do we need to coherently gather observations from multiple sensors, we also need to extract the salient information about an activity from a stream of imagery. Experience in recognition of geometric targets has shown that to ensure success we must build detailed predictive object models, which are used to reject clutter, resist noise and defeat occlusion. Models allow for the pooling of a large number of weak and noisy measurements into a coherent reliable estimate. Similarly, detailed *activity* models will be necessary for the recognition of activities. Unfortunately, activities are often more complex than single objects, involving a number of ambiguous objects, moving in distinctly non-rigid ways.

Take for example a platoon moving through a wood. One model of this activity is that 10 to 20 human figures must be moving in roughly the same direction for some period of time. Movement and heat[1] would help identify each of the primitive elements of the activity (the people). Each figure must be tracked and some attempt must be made to recognize it as human or animal. While it may be difficult to conclude with high confidence that all of the figures are human, models of group activity will allow

[1] While our methods should extend to sensors such as IR, we will focus only on video sensors.

us to propagate the label from one primitive object to others. A good estimate of force count can then be reported.

In this framework, recognition is a two way process. Activity models are used to coordinate and disambiguate information at lower levels. At the same time, lower level primitive information is used to select hypothetical models. This is similar to the alignment approach in computer vision, where a few edges taken together can be used to select a model. This model can then be rapidly verified by looking for evidence in the image.

3 Specific Research issues

Our goal is to use AVMs as a basis for monitoring activities; to support perimeter patrol, visual mines, urban security or other surveillance situations in which a set of distributed monitors must cooperate to detect and follow activities. The main components needed are: techniques for seamlessly fusing visual information observed at different times from different locations; and a framework in which to construct activity models so that activity can be reliably and efficiently detected. Below we sketch our intended approach to these problems.

Inexpensive, Robust AVMs: Our primary focus will be on *activity recognition* and *activity calibration*, and initially we will work with conventional cameras and workstations. We will also leverage earlier work on on small, active, eye-head systems. Of particular relevance to AVMs is our Cheap Vision Machine (CVM), an inexpensive, robust, compact vision module based on a C32 Digital Signal Processing chip[7, 6, 5]. The CVM in its typical configuration is coupled with a small camera, and in many instances also can pan and/or tilt the camera. Such a basic system is very useful for controlled observation and monitoring. In addition, we have implemented a suite of software modules that run in real-time on this platform, including simple edge detection, optical flow, motion tracking, and stereo vision. We have fabricated 15 CVMs, and have incorporated some of them into small mobile robots, executing real-time obstacle detection and avoidance[10], and map construction and navigation, among other computations. We have also used a CVM as a central component of an active eye/head system which can detect motion, foveate it and then track the detected object in real time (see Figure 1)[18]. Separately we have produced an inexpensive real-time trinocular stereo system, and which can be incorporated with the CVM into a prototype AVM.

Given an AVM testbed, we can design and implement a distributed system for activity detection and monitoring. We envision several components: **Calibration of the forest** – both geometric and activity calibration; **Primitive detection** – spatial detection of primitives, e.g. people, hands, faces, vehicles, and temporal detection of patterns of activity involving such spatial primitives; **Site models** – for establishing context to aid in interpretation of activities; **Hierarchical models of activity** – involving coordinated activities of multiple primitives, patterns of activity across temporal sequences.

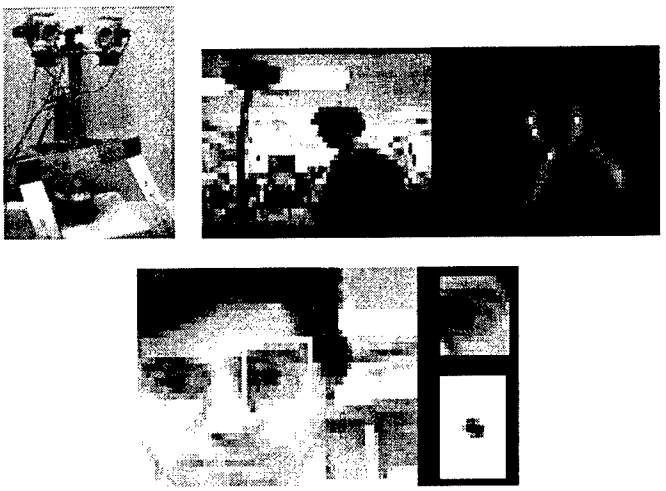


Figure 1: (top) A small, robust inexpensive tracking system. (middle) The system automatically detects motion and foveates. (bottom) Salient features are extracted and automatically tracked at video rates.

Calibration of the forest: To intelligently interpret data acquired from a distributed set of sensors, it is essential that we estimate their geometry. This means we need to know the location and orientation of each sensor, relative to a world coordinate frame, or relative to one another. to do this, we can use several sources of information[12, 11, 13, 14, 2, 3]: views from the cameras during placement, GPS for rough location, correspondence between static features, dynamic feature correspondences (i.e. those arising from moving objects), recovery of the affine, Euclidean and porjective geomotry of the forest of sensors using: 3D parallel lines in the scene provide information about the affine structure of the scene; 3D translational motions of rigid objects such as vehicles also provide the same sort of information; known distances, heights, angles or ratios of lengths provide information about the Euclidean geometry of the scene; 3D nontranslational motions (i.e. with a rotation component).

We will develop a complete system that combines some or all of the above features to obtain the best calibration possible, together with an estimate of its level (projective, affine, Euclidean, intermediate)

and of its quality. The system will have to decide which information it needs, which information it can use, and which information it is lacking. It should have a model of its current state to report to the user and should be able to send requests to this user for information that would allow it to improve its state. Such a calibration system will enable us to relate the geometry of different sensors to one another and to a common whole, a key precursor to detecting and classifying activities.

Primitives detection: To detect moving objects reliably, especially in a coordinated manner from multiple sensors, we need to bring several fundamental tools to bear on this problem.

First, normal optical flow, (along the image intensity gradient, will be utilized. Here we gain a great deal in simplicity of computation, although the information we collect from the images is less rich than with full optical flow. This, however, is where we can use the fact that the forest of sensors is actually a multi-camera stereo system. By having the sensors collaborate (or in vision terms by exploiting the redundancy between motion and stereo) we can actually use only the normal flow to derive interesting 3D properties of the moving object.

Second, tracking using optical flow and analysis of active contours will be used. Currently, such methods are used to track single non-overlapping objects, while we require the capability to track multiple, sometimes occluding or self-occluding objects. A recent advance in the implementation of the theory of nonlinear PDEs is the Osher-Sethian algorithms which can deal with evolving planar curves or surfaces whose topology may change over time. This is one of the aspects we need in order to solve the difficult task of tracking groups of people and vehicles. We propose to extend the recent concept of geodesic snakes[1] which have been developed for static images to the case of dynamic multiple images in order to track simultaneously in several views and to be able to cope with arbitrary changes of the topology of the tracked silhouettes. The fact that several images are available is of immediate benefit since tracking can use transfer techniques between images (possible due to calibration), thereby simplifying the underlying PDEs that drive the active contours.

Once we have extracted basic primitives of moving objects, we can turn to extracting useful information about those objects. This will involve both extracting information directly from the object (i.e. static and dynamic models of moving objects) and using general information about the scene (i.e. static models of the site to be monitored).

Site Modeling: While some interpretation of activities can occur by simply combining observations from multiple views of moving objects, static site models can also support activity detection and classification.

Figure 2: Given a current adaptive background image (UL) we can automatically do change detection (UR) and estimate the height of the objects (LL). This allows us to accumulate a depth image.

We believe that site models at different resolutions and accuracy can all contribute to this process: having very detailed, accurate models will clearly be of aid, but having approximate models can also be useful. In the latter category, we plan to develop a qualitative system that determines the rough spatial relationships of objects present in the scene. This system is entirely passive, and will rely on the tracking of moving objects to determine occlusion boundaries and relative depths. Models can be built from a video sequence recorded from a single stationary camera, or by combining views from multiple, roughly calibrated cameras. These models can thus be constructed to be consistent from multiple views. An example of this is shown in Figure 2 and Figure 3.

More detailed site models can be built by relying on the calibration of the forest of sensors[2, 3, 4,

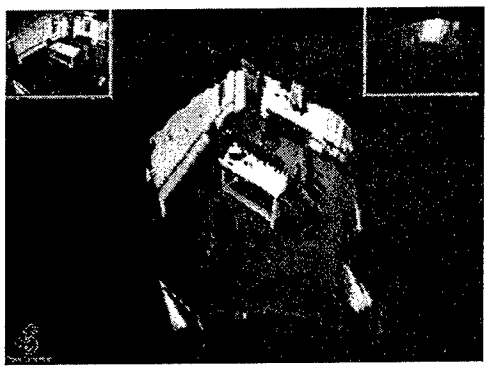


Figure 3: By tracking moving objects over time we can build up depth maps so that current views can then be mapped into a coarse 3D model.

15]. Depending on the level at which the calibration has been achieved, projective, affine or Euclidean reconstructions can be obtained.

A variety of potentially useful 3D reconstructions may then be produced ranging from sets of raw 3D data points reconstructed by intersecting optical rays from the sensors, to polyhedral or higher-order spline approximations of these points. These approximations can be built in several ways depending upon the context. A bottom-up approach would look for planar faces in the environment and then try to connect them into polyhedra using edge information. The planar faces can be detected in the images by using the fact that they induce collineations (linear projective transformations between those images; they can usually be well approximated by affine transformations). A top-down approach would use some parametric models of buildings, or parts thereof and would match them either directly on the raw 3D reconstruction or, preferably, onto the images themselves. A combination of the bottom-up and top-down approaches would probably increase speed and robustness. The 3D site model could then be labeled as, e.g., buildings, roads, trees.

Moving object modeling: In conjunction with the site models, the results of tracking can be used to derive detailed descriptions of moving objects. We will use the dynamics of the detected motion to distinguish vehicles from people from animals. This will be achieved in part by projecting parameterized models (which can be thought of as a parametric 3D snake) into the images where objects have been detected, and matching to the outlines of the hypothesized object simultaneously in all those images. Convergence of the 3D snake will provide acceptance or rejection of the hypothesis and, in the case of acceptance, information about the 3D shape, size and velocity of the the object. Vehicles will involve rigid motions and simples models. People will involve non-rigid, hierarchical, biomechanical models.

Activity calibration: An important issue related to the extraction of object models is the calibration of observations between sensors. Knowing the geometric relationship between sensors is an important first step, but we need to extend this to deal with observations of simultaneous activities, which will act to further refine these models. While this approach is related to camera calibration, it will be necessary to know more than the geometric relationship between the cameras. We propose to learn, through experience, how the activities in one image appear in another. We call this *Activity Calibration*. It is closely related to the Bayesian fusion of information from different noisy sensors. Activity calibration is a process by which the sensors learn to agree on their classifications.

There are multiple levels of detail in which activity calibration can occur. The simplest level involves coordination of simple motion between multiple sensors – relating the position and direction of a single moving object between views. The next level of detail is to use that simple calibration to reinforce the interpretation of actions between views. Our system will learn, through experience, how to use such multiple views to optimize classification of actions by utilizing the most salient views, and by combining information from multiple views to increase confidence in the classification. A related issue is to use the dynamical signatures themselves, as extracted from each sensor, to help establish correspondence of people/animals in multiple views. The Mutual Information method[17] is a strong candidate for establishing and testing these correspondences. Further up the hierarchy of actions would be those involving multiple moving objects, in which the same issue of coordinating interpretation of the multiply observed actions between the views will occur.

Learned Activity Models:

Given the ability to coordinate and calibrate both multiple sensors and the activities recorded by them, we can then consider the problem of modeling and classifying activities. We propose to construct dynamical models for the activities of interest. Models include primitive elements, and relationships between them in space, and more importantly in time. These models will be initialized by hand, but are then adjusted automatically to accurately match real situations. We believe that learning through repeated observation will be a critical component of model building. Learning will tune initially rough models so that they are accurate and predictive in their domain of application.

To create such models, we must first detect primitive elements. Every activity is a temporal sequence of actions performed by a number of primitive elements: soldiers, jeeps, tanks, etc. These must be detected and provisionally labeled by class. They must also be tracked. These primitives will play the same role as an edge does when recognizing a geometric object. Primitives need not be detected nor labeled with complete accuracy. The activity model will be used to correct or discard incorrect primitives.

To detect the primitive elements, we must consider both spatial and temporal detection. For example, we need generic methods that can detect people, largely independent of position and view; and similarly, we need generic methods that can detect vehicles, independent of position, view and type of ve-

hicle. We plan to investigate two approaches: one using flexible templates [16, 9], and the other using linear combinations of models[8]. Both methods will need extensions to deal with temporal coherence, and model deformations that evolve over time.

Thus, two of the approaches that we intend to explore are: (1) Encoding the ordinal photometric structure at different image regions at discrete time instants. This, in essence, would represent how the qualitative neighborhoods of different image regions are changing over time. (2) Qualitatively encoding the trajectories of different image regions. Interframe matching may be accomplished with an ordinal correlation technique (say, rank-order correlation). Furthermore, rather than storing precise optical flow maps, the flow vectors are assigned categorical attributes.

In summary, our goal is to use a forest of AVMs as a basis for monitoring activities; to support perimeter patrol, visual mines, urban security or other surveillance situations in which a set of distributed monitors must cooperate to detect and follow activities. We will attack this problem with several key components : **Calibration of the forest** – both geometric and activity calibration; **Primitive detection** – spatial detection of primitives, e.g. people, hands, faces, vehicles, etc., and temporal detection of patterns of activity involving such spatial primitives; **Site models** – for establishing context to aid in interpretation of activities; **Hierarchical models of activity** – involving coordinated activities of multiple primitives, patterns of activity across temporal sequences, etc.

References

[1] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Fifth ICCV*, pages 694–699, 1995.

[2] O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint.* MIT Press, 1993.

[3] O. Faugeras. Stratification of 3-d vision: projective, affine, and metric representations. *JOSA*, 12:465–484, 1995.

[4] O. Faugeras, S. Laveau, L. Robert, C. Zeller, and G. Csurka. 3-d reconstruction of urban scenes from sequences of images. In *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pages 145–168, 1995.

[5] I. Horswill. Integrating vision and natural language without central models. In *AAAI Fall Symposium on Embodied Language and Action*, 1995.

[6] I. Horswill. Visual routines and visual search: A real-time implementation and an automata-theoretic analysis. In *Proc. 14th IJCAI*, 1995.

[7] I. Horswill. Visual architecture and cognitive architecture. *Journal of Theoretical and Experimental Artificial Intelligence*, 1997.

[8] M. Jones and T. Poggio. Model-based mathcing of line drawings by linear combinations of prototypes. In *Fifth ICCV*, pages 531–536, 1995.

[9] P. Lipson, E. Grimson, and P. Sinha. Configuration based scene classification. In *CVPR*, 1997.

[10] L. Lorigo. Visually-guided obstacle avoidance in unstructured environments. Master's thesis, MIT, 1996.

[11] Q.-T. Luong and O. Faugeras. Camera calibration, scene motion and structure recovery from point correspondences and fundamental matrices. *IJCV*, 1996.

[12] Q.-T. Luong and O. Faugeras. The fundamental matrix: Theory, algorithms and stability analysis. *IJCV*, 17:43–76, 1996.

[13] Q.-T. Luong, Z. Zhang, and O. Faugeras. Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction. *IEEE Trans. Rob. Autom.*, 12:103–113, 1996.

[14] L. Robert and O. Faugeras. Relative 3-D positioning and 3-D convex hull computation from a weakly calibrated stereo pair. *Image and Vision Computing*, 13, 1995.

[15] B. Seales and O. Faugeras. Building three-dimensional object models from image sequences. *CVIU*, 61:308–324, 1995.

[16] P. Sinha. Object recognition via image invariants. *Investigative Ophthalmology and Visual Science*, 35, 1994.

[17] P. Viola and W. Wells. Alignment by maximization of mutual information. In *Fifth ICCV*, 1995.

[18] M. Wessler. An extensible visual tracking system. Master's thesis, MIT, 1995.

Robust Video Motion Detection and Event Recognition

Bruce Flinchbaugh

Texas Instruments
Research & Development
P.O. Box 655303, MS 8374
Dallas, Texas 75265
flinchbaugh@ti.com

Abstract

This report summarizes recent progress in video event recognition technology for automatically monitoring scenes, and outlines objectives of new research to improve reliability and extend the functionality. TI has demonstrated an event recognition capability that automatically processes video data at 10-20 frames per second and reports the events as they occur during long periods of observation. For example, as people, vehicles and objects move in the field of view, the system recognizes when entities enter and exit the scene, when a person deposits an object, when overall imaging conditions change, and when someone loiters in a specified area. The system has been demonstrated using an infrared video camera in darkness and CCD cameras in lighted areas. Ongoing research is enhancing the reliability of video motion analysis methods for robust performance in outdoor environments, and extending event recognition functionality for new kinds of events. This research will enable networked smart cameras for autonomous situational awareness of site perimeters, battlefields and other urban and rural areas where physical security and safety are primary concerns.

1 Research Objectives

The overall objective of this research is to develop and demonstrate new video processing methods for automatically monitoring scenes. Whereas cameras of today deliver images and video data, smart cameras of the future will deliver information derived from video data. These smart cameras will communicate via local and wide area networks to enable many new capabilities. For defense needs, smart cameras will autonomously deliver information about live events to distributed information systems that support battlefield awareness in urban and rural environments. Smart cameras will effectively extend the sight of commanders to remote areas by accurately drawing attention to important events in progress.

Specific goals are to develop video surveillance and monitoring methods to recognize new kinds of events, to improve the reliability of the moving object analysis process, and to demonstrate effectiveness of the new methods in performing important tasks. New event recognition methods will classify motions and interactions of objects into custom categories that are important for mission-specific tasks. Robust moving object detection and tracking is needed to interpret significant changes in video sequences as entities move in the field of view, especially amidst video changes caused by variations in illumination, temperature, wind, and occlusions.

2 Demonstration and Evaluation

Proof-of-concept demonstrations will emphasize physical security monitoring tasks in and around urban area buildings. The outdoor experiments will be of particular importance for battlefield awareness. For example, the infrared image of Figure 1

The research described in this report is sponsored in part by the DARPA Image Understanding Program.

More information about this research is available at:
http://www.ti.com/research/docs/iuba/index.html

shows a rural monitoring scenario in which a person has emerged from behind a tree and is walking across a grassy area. Exemplary tasks in this scenario are to reliably determine when a person is in the field of view, and to count the number of people who cross the field. To achieve practical demonstration goals, a variety of open-ended research issues must be resolved to some extent. What kinds of events can be recognized using a single video camera? What contextual information is needed for reliable video monitoring in a given situation? This research will contribute new insight while developing new functionality for smart cameras of the future.

Realistic video monitoring tasks will be used to test new techniques for robust moving object detection and event recognition, with two kinds of metrics for evaluating progress. Physical security monitoring experts will be consulted to select worthwhile new events to recognize, and to provide feedback about the quality of system performance compared to current practice. This evaluation will identify operational advantages of autonomous video event recognition systems. The primary quantitative metrics for characterizing performance are the error rates of event recognition reports. For example, if the task is to capture a single frontal view image of each person who loiters in a specified area, then non-frontal images, extra frontal images, and no frontal image of a loitering person would contribute to the error rate.

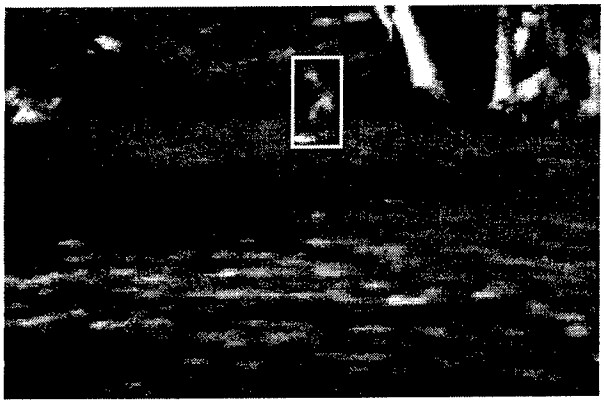


Figure 1. Autonomous video monitoring of remote areas draws attention to important events in progress.

3 Autonomous Video Surveillance Progress

In previous TI research [Flinchbaugh and Olson, 1996], several video monitoring techniques were devised to demonstrate feasibility of tracking people and marking their positions on a map display [Flinchbaugh and Bannon 1994], recognizing whether a person is holding a box [Rao and Sarwal, 1996], and recognizing some basic actions or events (enter, exit, deposit, remove, move, rest) of people and objects in the field of view [Courtney, 1997].

During the past year, an Autonomous Video Surveillance (AVS) system [Olson and Brill, 1997] has been developed that integrates the previous techniques for the first time, and provides several new integrated capabilities to monitor TV and infrared video cameras:

Calibration-Free Image-to-World Mapping: After an operator specifies approximate correspondences between selected image regions and map regions, the system estimates 3D locations of objects in the field of view without solving for the camera projection matrix or internal calibration parameters.

User Interface for Multiple Cameras: The map-based user interface has been extended to operate as a server for multiple video processors, allowing the operator to visually monitor tracks and event reports from multiple cameras, as positions of people and objects are dynamically plotted on a map.

Object Analysis: The system classifies objects that have been deposited in a scene as one of several known object types (e.g., box, briefcase, and notebook) or as an unknown object.

Contextual Alarms: The alarm monitoring system allows alarms to be conditioned on type of event, location, time of day, and the type of object involved in the event.

Best-View Selection: This method assesses the relative quality of two views of a person in a video sequence. This allows a video monitoring system to select and save a single high-quality digital snapshot of each person that enters the field of view.

Real-Time Operation Without Special Hardware: All of the above capabilities except object analysis run at 10-20 frames per second on a conventional workstation. This capability enables long-term experiments that were previously not feasible, and improves tracking and event recognition reliability.

The AVS system has been used to demonstrate feasibility of generating real-time alarms for specified events in three security monitoring scenarios. These demonstrations illustrate how physical security can be partially automated to monitor hallway, office, and building perimeter areas. In each area, a camera provides live video data of scenes in the field of view, while the AVS system monitors the video to analyze events and signal alarms.

Hallway Monitoring: Consider the scenario illustrated in Figure 2. The AVS system detects and tracks people as they walk in office building hallways. Alarms are interactively defined for conditions such as when someone loiters in a specified area or enters a particular office. Autonomous visual assessment provides information to augment other information, such as biometric access control information at building entrance points.

Figure 2. In a hallway monitoring demonstration, the AVS system tracks people and signals an alarm when someone loiters in a specified area.

Room Monitoring: For the room monitoring scenario shown in Figure 3, the AVS system maintains a situational awareness record of events and signals alarms for a variety of specified conditions. For example, an alarm may be specified for events in which a person places a briefcase on a table, but not if the person leaves a box on the floor. Using contextual information such as time of day and access control identification, the system can report other alarm conditions that are functions of who is in the room and when.

Perimeter Monitoring: For perimeter monitoring scenarios, an infrared camera is used in a dark area to provide video data to AVS, illustrating the ability to monitor areas outside buildings at night. For example, the AVS system could monitor a building entrance and signal an alarm if someone walks by and leaves an object outside the door, as illustrated in Figure 4), but not if someone loiters without placing an object on the ground.

Figure 3. Automatic room monitoring provides concise reports of activities in the field of view

Figure 4. An outdoor site perimeter surveillance scenario involves an infrared video camera to recognize events in darkness

Acknowledgments

Frank Brill and Tom Olson developed the new capabilities described in this report.

References

[Courtney, 1997] J. D. Courtney. Automatic video indexing via object motion analysis. *Pattern Recognition*, **30**(4), April 1997.

[Flinchbaugh and Bannon, 1994] B. Flinchbaugh, and T. Bannon. Autonomous scene monitoring system. In *Proc. 10th Annual Joint Government-Industry Security Tech. Symposium*, American Defense Preparedness Association, June 1994.

[Flinchbaugh and Olson, 1996] B. Flinchbaugh and T. Olson. Autonomous video surveillance. In *Proceedings of the SPIE: Emerging Applications of Computer Vision*, D. Schaefer and E. Williams, editors, Washington, D.C., vol. **2962**, pp. 144-153, October 1996.

[Olson and Brill, 1997] T. J. Olson and F. Z. Brill. Moving object detection and event recognition for smart cameras. In *1997 Proceedings of the DARPA Image Understanding Workshop,* Morgan-Kaufman, May 1997.

[Rao and Sarwal, 1996] K. Rao, and P. Sarwal. A computer vision system to detect 3-D rectangular objects. In *Proceedings of the Third IEEE Workshop on Applications of Computer Vision*, pp. 27-32, 1996.

Omnidirectional VSAM Systems: PI Report *

Shree K. Nayar† and Terrance E. Boult‡

†Dept. of Computer Science, Columbia University, New York
Email: nayar@cs.columbia.edu, URL: http://www.cs.columbia.edu/nayar

‡Dept. of Elec. Engg. and Comp. Sc., Lehigh University, Bethlehem, PA
Email: tboult@eecs.lehigh.edu, URL: http://www.eecs.lehigh.edu/boult

Abstract

Traditional video surveillance and monitoring (VSAM) systems rely on off-the-shelf lenses and video cameras that provide limited fields of view. This research program is geared towards the development and application of catadioptric video cameras that have unusually large fields of view. Our recent work has demonstrated the use of catadioptric image formation to achieve a truly omnidirectional video camera. In this report, we begin by summarizing our results and then proceed to outline our plan for future work.

1 Introduction

Today's video surveillance and monitoring (VSAM) platforms rely heavily on conventional imaging systems as sources of visual information. Any conventional imaging system is limited in its field of view. It is only capable of acquiring visual information through a relatively small solid angle subtended in front of the image detector. It might appear that this field of view problem can be resolved by simply using a large number of conventional cameras that are packed close together, each pointing in a different direction. This is better than not seeing in some directions, but such an approach has two fundamental problems. First, it cannot be compact in hardware and processing. Second, and more seriously, the cameras cannot share the same *center of projection* (viewpoint) which is a geometrical requirement that serves as the basis for a large body of work in image understanding.

Imagine that we had at our disposal an image sensor that could "see" in *all* directions from its location (single viewpoint) in space, i.e. the entire "sphere of view". We refer to such a sensor as being *omnidirectional*. It is easy to see that this hypothetical device would have a profound impact on the nature of VSAM systems and their capabilities. (a) An omnidirectional sensor would allow the VSAM system to be, at all times, aware of its *complete* surrounding. (b) Tracking a moving object would be feasible in software, *without* the need for any moving parts (i.e. no panning, tilting, and rotating). (c) Unlike physically directed cameras, the omnidirectional sensor would have no problem *simultaneously* detecting multiple objects (or intruders) in distinctly different parts of its environment.

Is it feasible then to turn our hypothetical omnidirectional sensor into reality? The answer is indeed in the affirmative. This claim is based on our very recent result [Nayar, 1997] on the use of reflecting surfaces (mirrors) to enhance the field of view of conventional imaging systems. We have shown that none of the existing wide-angle image sensors are omnidirectional in the true sense of the word. Our analysis has led to an omnidirectional sensor that has a hemispherical field of view while maintaining a single viewpoint [Nayar, 1997]. Two such sensors can be placed back-to-back to acquire the entire sphere of view. In addition, our theoretical analysis has revealed the entire class of omnidirectional cameras that can be realized using mirrors [Nayar and Baker, 1997]. Finally, we have developed a real-time software system that can generate scores of perspective video streams (for user selected viewing parameters) from a single omnidirectional video stream, using no more than a PC [Peri and Nayar, 1997].

In this report, we summarize our research results on omnidirectional image sensing and detail our plans for future work. The intended goals of this research program are: (a) compact, low-cost, omnidirectional sensors that can serve as the basis for future VSAM platforms, (b) real-time software systems that map sections of an omnidirectional image to high-resolution perspective, or panoramic views, (c) algorithms to fuse image stream data (either omnidirectional or traditional) to yield super-resolution data, (d) fast algorithms for object detection and activity recognition, (e) omnidirectional approaches to egomotion estimation and image stabilization, (f) strategies for developing a cooperating net-

*The prior work summarized in this report was supported in parts by the DARPA/ONR MURI Grant N00014-95-1-0601, an NSF National Young Investigator Award, and a David and Lucile Packard Fellowship. The planned research will be supported by a DARPA contract awarded in response to BAA 96-14.

work of distributed omnidirectional sensors, (g) collaboration with the VSAM IFD team to explore novel applications of omnidirectional sensors to real-time surveillance, monitoring, tracking, and recognition, (h) an extension of the IUE sensor hierarchy to handle conventional and omnidirectional image streams, and (i) an extension of (parts of) the IUE to be CORBA compliant and support distributed VSAM algorithms.

2 Approaches to Wide-Angle Imaging

We first review the state of the art in wide field-of-view image sensing. Detailed surveys can be found in [Nalwa, 1996] and [Nayar, 1997]. Existing wide-angle imaging systems can be broadly classified into the following three categories.

Rotating Imaging Systems: An obvious solution is to rotate a traditional imaging system about its center of projection. The sequence of images acquired by rotation are "stitched" together to obtain a panoramic view of the scene. Such an approach has been recently proposed by a few investigators (see [Chen, 1995], [McMillan and Bishop, 1995], [Krishnan and Ahuja, 1993]). The first disadvantage of this approach is that it requires the use of moving parts and precise positioning. The more serious drawback lies in the total time required to obtain an image with enhanced field of view; this makes the approach impractical for real-time applications.

Fish-Eye Lenses: An interesting approach to wide-angle imaging is based on the fish-eye lens (see [Oh and Hall, 1987]). Such a lens is used in place of a conventional camera lens and has a very short focal length that enables the camera to view objects within as much as a hemisphere. While this works reasonably well, fish-eye lens do not ensures that the viewpoint of the imaging system is fixed for all points in the scene, thus precluding their use in many traditional IU algorithms. Further, to view an entire hemisphere, the fish-eye lens must be quite large and hence expensive.

Reflecting Surfaces: This class of imaging systems use reflecting surfaces (mirrors) to enhance the field of view. The shape, position, and orientation of the reflecting surface are related to the viewpoint and field of view in a complex manner. While it is easy to construct a configuration which includes one or more mirrors that dramatically increases the field of view of the imaging system, it is hard to keep the effective viewpoint fixed in space. A few different mirror shapes and configurations have been suggested in the past [Yagi and Kawato, 1990] [Hong, 1991] [Yamazawa *et al.*, 1995] [Nalwa, 1996]. Two features of existing implementations are worth noting. (a) Those based on planar mirrors require multiple imaging devices and digitizers [Nalwa, 1996]. (b) Those based on curved mirrors (except for the one in [Yamazawa *et al.*, 1995]) violate the fixed viewpoint constraint; the acquired image cannot be used to compute perspective images that are geometrically consistent.

3 Summary of Research Results

Our approach to wide-angle imaging lies in the last of the abovementioned categories; we incorporate reflecting surfaces (mirrors) into conventional imaging systems. This is what we refer to as *catadioptric*[1] image formation. As recently noted in [Yamazawa *et al.*, 1995] and [Nalwa, 1996], the resulting imaging system must have a single center of projection (viewpoint). We show that under orthographic projection the curved mirror that produces a single viewpoint is parabolic [Nayar, 1997]. Precise orthographic projection is indeed feasible [Watanabe and Nayar, 1996] and makes the mapping from the acquired omnidirectional image to perspective images straightforward from a computational perspective. Our analysis has led us to what we view as a practical omnidirectional video camera [Nayar, 1997].

We have implemented several prototypes of the proposed camera, each one designed to meet the requirements of a specific application. The single viewpoint constraint allows us to compute, distortion-free (perspective) images of the scene for any user-selected viewing direction, focal length and image size [Nayar, 1997] (see Figure 1). In [Peri and Nayar, 1997], a software system is described that generates a large number of perspective and panoramic video streams from a single omnidirectional video input. In addition, we have derived a complete class of solutions for catadioptric image formation [Nayar and Baker, 1997] that satisfy the single viewpoint constraints. This general solution allows us to evaluate the merits and drawbacks of specific mirror shapes, including ones proposed in the past.

[1] *Dioptrics* is the optics of refracting elements (say, lenses) whereas *catoptrics* is the optics of reflecting surfaces (mirrors) [Hecht and Zajac, 1974]. The combination of refracting and reflecting elements is referred to as *catadioptrics*.

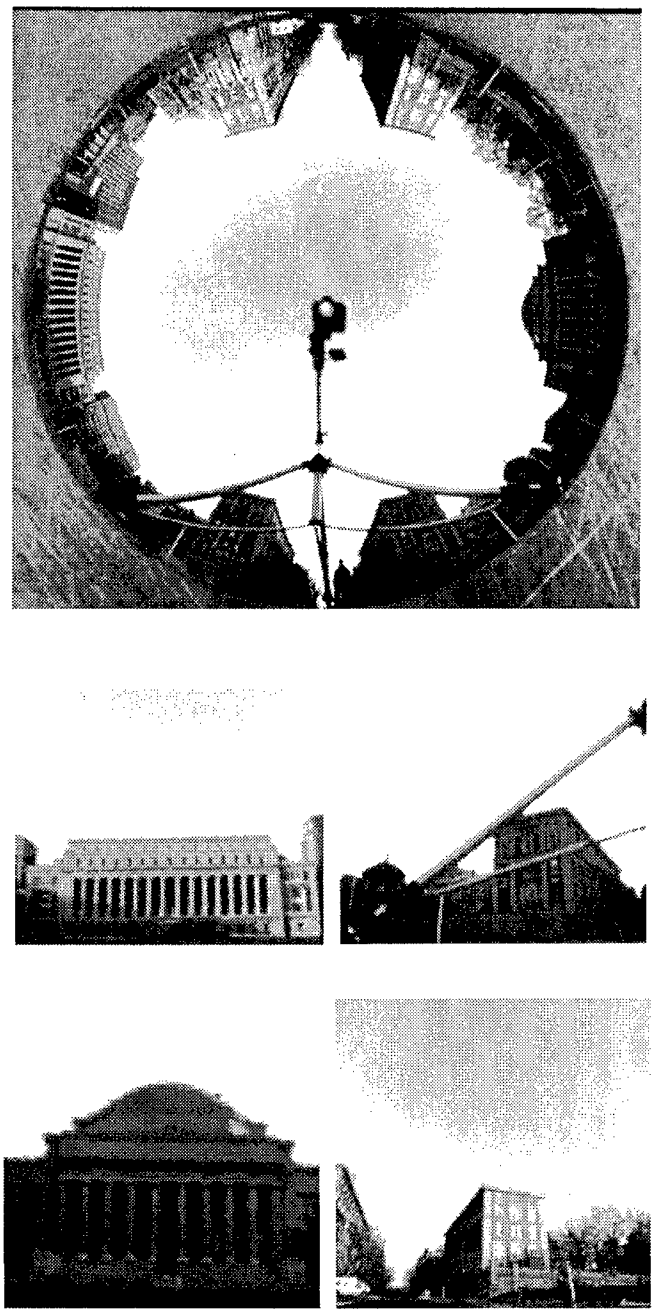


Figure 1: Software generation of perspective images (bottom) from an omnidirectional image (top). Each perspective image is generated using user-selected parameters, including, viewing direction (line of sight from the viewpoint to the center of the desired image), effective focal length (distance of the perspective image plane from the viewpoint of the sensor), and image size (number of desired pixels in each of the two dimensions). It can be seen that the computed images are indeed perspective; for instance, straight lines are seen to appear as straight lines though they appear as curved lines in the omnidirectional image.

4 Research Plan

We now outline our future work and describe the enabling technologies that will be delivered. We have partitioned our work into several tasks. Multiple tasks will be executed simultaneously.

4.1 Theoretical Model for Catadioptric Image Formation

As stated earlier, we have completed the development of a few different prototypes of the omnidirectional camera [Nayar, 1997]. In order to best exploit the images produced by such a sensor, an advanced model of image formation is needed. Unlike the linear perspective lenses that are commonly used, our omnidirectional cameras are catadioptric systems that use a combination of reflecting elements (mirrors) and refracting elements (lenses). Such a system can be viewed as a cascaded (relayed) optical system, where the mirror plays the role of a somewhat unconventional lens. Using such a modeling procedure, we will characterize the following parameters of the system: focal length, depth of field, and optical resolution in the image plane. In particular, optical resolution poses an interesting problem. Since the mirror is curved, the effects of coma, astigmatism and chromatic aberration need to be studied [Born and Wolf, 1965]. The end result of this analysis will be a sophisticated, modular image formation model. The components of this model will serve as the basis for accurate generation and analysis of computed perspective and panoramic images.

4.2 Generation of High-Quality Novel Views

We already have a working implementation of a software system that creates several perspective and panoramic images at video-rate from a single omnidirectional video stream [Peri and Nayar, 1997]. The system called OmniVideo uses several advanced software design techniques to generate multiple perspective and panoramic video ports using no more than a digitizer and a Pentium Pro PC. The viewing parameters (viewing direction, magnification and image size) are selected interactively by the user via an input device such as a joystick. The present implementation simply maps points in the omnidirectional image to points in any desired perspective or panoramic image (see [Peri and Nayar, 1997]).

Our goal is to use the image formation model described in section 4.1 to significantly enhance the quality of the computed perspective images. This will be done in three stages. (a) We will use interpolation schemes that exploit the spatially varying mapping between omnidirectional and

perspective images. This is different from the traditional approach of linear shift-invariant interpolation [Mintzer and Braudaway, 1995]. (b) We will use our model for coma and astigmatism to deblur the images. This will require the development of a linear shift-variant filter [Sawchuck, 1974] that will sharpen the images without introducing aliasing artifacts. (c) The computed images will be further processed to enhance critical features using extensions to the techniques developed in [Boult and Wolberg, 1993]. Since our goal is real-time surveillance and monitoring, novel data structures will be employed to achieve the desired efficiency in image generation. This technology will enable the VSAM system to use high-quality pure perspective images of areas of activity. The tradeoffs between computational speed and image quality will be evaluated.

4.3 Evaluation of Omnidirectional Video Processing

The performance evaluation of the sensor and the supporting software system will be conducted at two sites, Columbia University and Lehigh University. In the first phase, we will evaluate the quality of computed perspective images by studying the image quality of known features such as lines and corners. One objective here is to determine the quality of the geometric mapping and its adherence to linear perspective projection. This will be done by computing lines in the perspective images and evaluating their linearity. In addition, features such as corners will be examined to estimate the quality of the interpolation and enhancement schemes outlined in section 4.2. The image structure in the area of the feature will reveal the accuracy of the computed images as well as the presence/absence of undesirable effects such as aliasing and blurring.

The second phase of our evaluation will be conducted at a vision task level. Here, the purpose is to estimate the performance of a high level vision task in the context of omnidirectional sensing. This evaluation will use the appearance-based matching system, SLAM[Nene *et al.*, 1994] (probably the IUE port thereof), in an off-line mode. A preprocessing step will extract regions of interest, defined as large regions moving with respect to the background. The system will be trained for static "recognition" using the perspective projection computed from the regions of activity. To keep the experiments reasonable in time/labor demands, the recognition database will be kept small (10-20 objects) but will include objects closely related to our intended applications, e.g. various (plastic models of) ground/air vehicles. Errors in both recognition rate and pose estimation will serve as our performance metrics.

4.4 Demonstration of the Omnidirectional System to the VSAM IFD Team

Subsequent to performance evaluation, we plan to demonstrate a complete system (prototype sensor and supporting software) to the VSAM IFD team to obtain feedback that we can use to maximize the applicability of our results. The process of interacting with the IFD team will continue throughout the proposed research.

4.5 Algorithms for Omnidirectional Flow Fields

Our goal is to address VSAM applications where the sensor is static as well as dynamic. In both cases, detection of motion is of utmost importance. To this end, we will study in detail the computation of optical flow fields from omnidirectional image sequences. At first thought, it may appear that standard flow computation methods may be directly applicable. Though rough estimates of flow are indeed obtainable using standard methods, accurate flow fields would require a more careful analysis. For instance, the optical flow constraint equation [Horn and Schunck, 1981] that is in wide use needs to be revisited in the context of omnidirectional imagery. This is because, under translation or rotation, a scene point not only shifts in the omnidirectional image but also distorts. The extent and type of distortion is solely dependent on the location of the point in the image.

Using our catadioptric image formation model, we will derive a new set of optical flow constraint equations. These equations will be applicable not only to the particular sensor we have developed but also others that seek to capture omnidirectional images, such as, fish-eye lenses and hyperboloidal mirrors. We expect our analysis to result in a novel framework for the computation of flow fields using unconventional sensors. Using this framework, we will develop algorithms for robust and precise optical flow estimation as well as more efficient ones for computation of normal flow fields.

4.6 Novel Algorithms for Egomotion

Of particular interest to mobile VSAM applications is the computation of egomotion and the detection of local object motions. All egomotion algorithms in use today rely on the data collected from small fields of view. In such cases, all of the 3D motion vectors lie "in front" of the sensor. This greatly limits the motion information contained in the acquired image sequences and hence restricts what can be computed from the sequences. Omnidirectional image sequences are

expected to have a profound impact on the way we view motion estimation. Since a single sensor is capable of capturing a hemispherical field of view, translation and rotation can be calculated with much higher robustness. Novel algorithms will be developed for egomotion based on the spherical representation of flow fields [Fermuller and Aloimonos, 1995], [Nelson and Aloimonos, 1988], [Yen and Huang, 1983]. Of particular interest to us is an algorithm that will estimate egomotion from normal flow rather than actual flow, since normal flow can be determined more efficiently.

4.7 Omnidirectional Image Stabilization

Present algorithms for motion detection [Irani and Anandan, 1996] and image stabilization [Yao *et al.*, 1996] are all based on the perspective projection model. Using our catadioptric models for omnidirectional image formation, we will pursue significant modifications to existing algorithms. Of particular interest will be stabilization algorithms that use horizon features; algorithms for which the omnidirectional sensors are particularly well suited. These algorithms are known to have theoretical advantages but have had limited utilization in the past because small field of view cameras are not guaranteed to see the horizon.

The developed optical flow, egomotion and image stabilization algorithms will enable a mobile VSAM system to automatically detect areas of activity (regions of interest) in the omnidirectional image. Each region of activity will then be mapped to a perspective image, scaled to a size suitable for visual analysis and enhanced using our interpolation and deblurring techniques. These algorithms will allow an omnidirectional VSAM system to detect and track multiple regions of interest that are in distinctly different parts of its large field of view.

4.8 Super-Resolution Techniques Based on Temporal Sequences

In the context of VSAM, there is an inherent tradeoff: field-of-view of the omnidirectional sensor versus resolution. We propose the development of advanced algorithms that extract high spatial and intensity resolution from multiple omnidirectional images. We propose fusing multiple frames with direct robust estimation techniques, such as, averaging the central quintile of image data or reweighted least squares. Various robust estimation techniques will be evaluated and compared with the previous approach. We also use more sophisticated image reconstruction techniques that include a model of sensor blur. The issue of sub-pixel registration, while well studied in the general context [Boult and Wolberg, 1993], also needs refinement for omnidirectional sensors.

A second version of the super-resolution algorithm will be developed with more explicit modeling of matching, sensing, and imaging "assumptions" and desired feature characteristics. It will fit the data using robust estimation (nonlinear M-estimation with dynamic reweighting). We will also develop a super-resolution approach which is less sensitive to view and lighting variations.

4.9 Omnidirectional VSAM Systems

Our super-resolution imaging, stabilization and tracking algorithms will allow us to robustly detect moving objects. Each time such an object is detected, it will be clipped out of the omnidirectional image, mapped into its pure perspective version, compressed and stored with a time stamp. A remote observer will be automatically notified when one or more regions of change are detected. This would permit a remote observer or an image analyst to focus attention on the regions of change. A graphical user interface will be developed that allows the user to interact with the omnidirectional image and regions of change. For instance, a stored region of interest can be zoomed into using the proposed perspective image generation software and enhanced in quality using the proposed super-resolution techniques. In addition, we wish to explore the application of appearance-based recognition algorithms [Murase and Nayar, 1995] [Nayar *et al.*, 1996] for identification of prestored objects, motions, and activities. We have already begun the development of algorithms for temporal appearance matching [Nayar *et al.*, 1996] that we expect will be more broadly applicable to activity detection than previous ones. The above algorithms will be developed with feedback from the VSAM IFD team.

4.10 Omnidirectional Platform with Active Head

Though our image enhancement and super-resolution algorithms will improve the quality of the detected images, we are bound to face VSAM applications where it is desirable to image regions of activity with very high resolution. For such cases, we propose a platform configuration that includes an omnidirectional sensor as well as a very fast active head. The active head will include a single high-resolution camera attached to a motorized zoom lens. The sole purpose of the head is to point and zoom into regions of activity detected by the omnidirectional sensor. The control scheme used to drive the platform will

be a *master-slave* one where the active camera is driven by commands from the omnidirectional one. This configuration is expected to have significant advantages. The omnidirectional sensor will ensure that the VSAM system is devoid of blind-spots. At the same time, the active camera captures regions of activity with the highest possible resolution and focus. The super-resolution algorithms will also be available to the active camera.

4.11 "Live" Image Sequences in the IUE

To make our sensors and algorithms readily accessible to users, we wish to use the IUE [Dolan *et al.*, 1996] [Mundy *et al.*, 1992] as the supporting environment. A significant problem to be addressed to support VSAM in the IUE is an extension to support "continuous" image sequences, bounded only by the amount of memory the programmer wants to allocate to them. We propose to extend the IUE sensor hierarchy, and then to implement a new class: *streaming-sensors* which will support both polling for new image and event-driven processing. The latter will require adding multi-threading to the IUE (a non-trivial extension). Because of current differences in operating support for multi-threading, the proposed work will take place only for the Solaris version of the IUE, with ports to other systems optional, depending on demand, funding and the OS support for lightweight threads.

4.12 Cooperating Distributed Sensors in the IUE

Distributed processing/communication among VSAM applications is going to be a necessary requirement for tomorrows battlefields and CORBA is emerging as a central component in distributed object-oriented applications. We will extend the IUE to have a CORBA interface and support the distributed computations need for the later years of our project. We will initially design and implement the CORBA interface needed for our omnidirectional sensor and VSAM applications. We will also design the interface for other IUE objects and, if resources permit, will implement them as well.

4.13 Infra-red Omnidirectional Sensors

An interesting problem we wish to explore is the development of omnidirectional sensors that use IR image detectors. Such a sensor would enable omnidirectional VSAM at low-light levels and during the night. The design of such a sensor poses several challenges. (a) The thermal energy needs to be reflected by the reflecting surface into the imaging device. This problem can be approached by using a reflecting surface whose temperature is held constant. (b) The more serious problem involves the effective focal length of existing thermal sensors. The optical parameters of IR sensors differ significantly from those of visible light sensors. We would like to investigate these differences and come up with the design of an IR omnidirectional sensor. If the design proves feasible, an IR sensor will be developed and evaluated.

5 Conclusion

This research project is geared towards the development of VSAM systems that are based on omnidirectional video cameras. Our initial results indicate that the our sensors can have a far-reaching impact on VSAM. Though our primary objective here is to advance the state-of-the-art in VSAM technology, our sensors can be exploited in almost any imaging application, including, automatic target recognition, real-time collection of wide-angle world maps, and autonomous navigation. As in our past work, technology transfer is of key interest to us. In the current project, this will be realized through the incorporation of our sensor models and temporal processing algorithms into the IUE.

References

[Born and Wolf, 1965] M. Born and E. Wolf. *Principles of Optics.* London:Permagon, 1965.

[Boult and Wolberg, 1993] T. Boult and G. Wolberg. Local Image Reconstruction and ubpixel Restoration Algorithms. *CVGIP: Graphical Models and Image Processing*, 55:63–77, Jan 1993.

[Chen, 1995] S. E. Chen. QuickTime VR - An Image Based Approach to Virtual Environment Navigation. *Computer Graphics: Proc. of SIGGRAPH 95*, pages 29–38, August 1995.

[Dolan *et al.*, 1996] J. Dolan, C. Kohl, R. Lerner, T. Boult, J. Mundy, and R. Bev eridge. Solving Diverse Image Understanding Problems using the Image Understanding Environment (IUE),. *Proc. of ARPA Image Understanding Workshop*, pages 1481–1504, Feb 1996.

[Fermuller and Aloimonos, 1995] C. Fermuller and Y. Aloimonos. Direct perception of three-dimensional motion from patterns of visual motion. *Science*, 270:1973–1976, 1995.

[Hecht and Zajac, 1974] E. Hecht and A. Zajac. *Optics*. Addison Wesley, Reading, Massachusetts, 1974.

[Hong, 1991] J. Hong. Image Based Homing. *Proc. of IEEE International Conference on Robotics and Automation*, May 1991.

[Horn and Schunck, 1981] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[Irani and Anandan, 1996] M. Irani and P. Anandan. A Unified Approach to Moving Object Detection in 2D and 3D Scenes. *Proc. of ARPA Image Understanding Workshop*, pages 707–718, Feb 1996.

[Krishnan and Ahuja, 1993] A. Krishnan and N. Ahuja. Range Estimation From Focus Using a Non-frontal Imaging Camera. *Proc. of the Eleventh National Conference on Artifical Intelligence (AAAI 93)*, pages 830–835, July 1993.

[McMillan and Bishop, 1995] L. McMillan and G. Bishop. Plenoptic Modeling: An Image-Based Rendering System. *Computer Graphics: Proc. of SIGGRAPH 95*, pages 39–46, August 1995.

[Mintzer and Braudaway, 1995] F. Mintzer and G. W. Braudaway. Processing Color Images While Preserving Color. Technical Report RC 20208 (89411), IBM Research Report, T. J. Watson Research Center, October 1995.

[Mundy *et al.*, 1992] J. Mundy, T. Binford, T.Boult, A. Hanson, R. Beveridge, R. Haralick, V. Ramesh, C. Kohl, D. Lawton, D. Morgan, and K. Price. The image understanding environments overview and class definitions. Technical report, Columbia Univ., Department of CS, and Lehigh Univ., Department of EECS, 1992. Document Coordinator: T. Boult. IUE Overview Document (300+ pages) and IUE Class Definitions (525 pages). Updated documentation at http://www.aai.com/AAI/IUE/IUE.html and http://www.eecs.lehigh.edu/IUE/IUE.html.

[Murase and Nayar, 1995] H. Murase and S. K. Nayar. Visual Learning and Recognition of 3D Objects from Appearance. *International Journal of Computer Vision*, 14(1):5–24, January 1995.

[Nalwa, 1996] V. Nalwa. A True Omnidirectional Viewer. Technical report, Bell Laboratories, Holmdel, NJ 07733, U.S.A., February 1996.

[Nayar and Baker, 1997] S. K. Nayar and S. Baker. Catadioptric Image Formation. *Proc. of DARPA Image Understanding Workshop*, May 1997.

[Nayar *et al.*, 1996] S. K. Nayar, H. Murase, and S. A. Nene. Parametric Appearance Representation. *Early Visual Learning, by S. K. Nayar and T. Poggio (editors)*, pages 131–163, April 1996. Oxford University Press, New York.

[Nayar, 1997] S. K. Nayar. Omnidirectional Video Camera. *Proc. of DARPA Image Understanding Workshop*, May 1997.

[Nelson and Aloimonos, 1988] R. C. Nelson and J. Aloimonos. Finding motion parameters from spherical motion fields (or the advantages of having eyes in the back of your head). *Biological Cybernetics*, 58:261–273, 1988.

[Nene *et al.*, 1994] S. A. Nene, S. K. Nayar, and H. Murase. SLAM: Software Library for Appearance Matching. *Proc. of ARPA Image Understanding Workshop*, November 1994.

[Oh and Hall, 1987] S. J. Oh and E. L. Hall. Guidance of a Mobile Robot using an Omnidirectional Vision Navigation System. *Proc. of the Society of Photo-Optical Instrumentation Engineers, SPIE*, 852:288–300, November 1987.

[Peri and Nayar, 1997] V. Peri and S. K. Nayar. Generation of Perspective and Panoramic Video from Omnidirectional Video. *Proc. of DARPA Image Understanding Workshop*, May 1997.

[Sawchuck, 1974] A. A. Sawchuck. Space-variant image restoration by coordinate transformation. *Journal of Optical Society of America*, 64(2):138–144, February 1974.

[Watanabe and Nayar, 1996] M. Watanabe and S. K. Nayar. Telecentric optics for computational vision. *Proc. of European Conference on Computer Vision*, April 1996.

[Yagi and Kawato, 1990] Y. Yagi and S. Kawato. Panoramic Scene Analysis with Conic Projection. *Proc. of International Conference on Robots and Systems (IROS)*, 1990.

[Yamazawa *et al.*, 1995] K. Yamazawa, Y. Yagi, and M. Yachida. Obstacle Avoidance with Omnidirectional Image Sensor HyperOmni Vision. *Proc. of IEEE International Conference on Robotics and Automation*, pages 1062–1067, May 1995.

[Yao *et al.*, 1996] Y. S. Yao, P. Burlina, and R. Chellappa. Stabilization of Images Acquired by Unmanned Groud Vehicle. *Proc. of ARPA Image Understanding Workshop*, pages 687–694, Feb 1996.

[Yen and Huang, 1983] B. Yen and T.S. Huang. Determining 3-d motion and structure of a rigid body using the spherical projection. *CVIP*, 21:21–32, 1983.

Image-Based Scene Rendering and Manipulation Research at the University of Wisconsin

Charles R. Dyer*
Department of Computer Science
University of Wisconsin
Madison, WI 53706
E-MAIL: dyer@cs.wisc.edu
HOMEPAGE: http://www.cs.wisc.edu/~dyer

Abstract

This report summarizes the research effort at the University of Wisconsin in support of the VSAM Program. Our primary goal is to develop technologies so a user can interactively visualize and virtually modify a 3D environment from a set of images. Current approaches are described for image-based scene rendering, scene manipulation, and appearance modeling.

1 Introduction

The ultimate goal of this project is to develop image-based methods that will enable a user to control the motion of a virtual camera so as to visualize a real 3D environment for applications such as facility monitoring and mission rehearsal. This technology will enable the rapid creation of an interactive visualization capability in which new views are synthesized by adaptively combining or "steering" a set of input images of the environment.

Our approach is image-based in the sense that all input and output about the scene is via images. Since the desired results are images this means that techniques should focus on producing realistic images, not feature space descriptions for classification, 3D model building, or other traditional computer vision goals.

The major challenge of this formulation is to create ways of combining a set of images, obtained from a fixed set of viewpoints, so that a user can interactively survey the real environment by controlling a virtual camera. To create a compelling sense of visual presence, the following user capabilities are key: (1) can interactively change viewpoint with respect to the 3D environment, (2) can render the environment photorealistically at high resolution, frame rate and quantization rate, and (3) can effect virtual changes in the acquired environment.

Our research activities are directed towards accomplishing these three capabilities of interactive virtual camera control, photorealistic rendering, and virtual scene modification operations. This report summarizes current activities related to these issues, emphasizing two new approaches to view synthesis. The first is called **view morphing** and treats primarily the two-input-view case. That is, given a pair of images of a static 3D scene, interpolate in-between views. The second approach treats the general case of synthesizing views over a wide range given an arbitrary number of input views, widely distributed around the environment. This **voxel coloring** approach reconstructs a photometrically-consistent volumetric representation of the scene, which is then used to render new views. The construction of this representation also allows the user to interactively modify the scene through image editing operations, and this capability, called **plenoptic image editing** (joint work with the University of Rochester), is also briefly summarized.

*The support of the Defense Advanced Research Projects Agency, and the National Science Foundation under Grant No. IRI-9530985 is gratefully acknowledged.

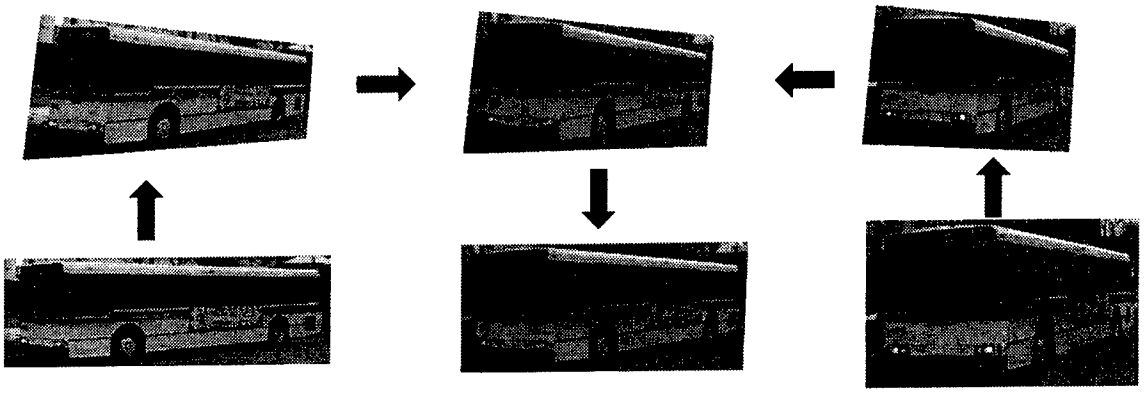


Figure 1: View morphing in three steps. Two original images (bottom left and right) of a bus are first prewarped (top left and right) to make the cameras parallel, and then morphed to create an in-between view (top middle). The desired gaze direction of the morphed view is established with a postwarp operation (bottom middle).

2 View Synthesis from Two Views

Recently, a number of methods have been developed by researchers in computer vision and computer graphics for synthesizing new views from one or more images. Our approach, called **view morphing** [Seitz and Dyer, 1996a, Seitz and Dyer, 1996b, Seitz, 1997, Seitz and Dyer, 1997c], builds on existing image morphing methods for creating compelling image sequences that smoothly transform one image into another. The method synthesizes images corresponding to new viewpoints by performing simple image warping operations. This work extends to perspective projection our earlier results [Seitz and Dyer, 1995].

The fundamental research questions associated with this area are, *when* and *how* can physically-correct new perspective views of a 3D scene be predicted from a set of basis views? With respect to the first question we have shown that when two basis images have the same scene points visible in each, a constraint we call *monotonicity*, this is sufficient to uniquely predict the appearance of the scene for all in-between viewpoints on the line segment connecting the input cameras' optical centers. This is true despite the fact that there may not be sufficient information in the pair of images to uniquely reconstruct the 3D scene. In other words, while any number of distinct scenes could have produced the given input images, the monotonicity constraint guarantees that each of those scenes will project to the same set of in-between images.

To answer the "how" question we have developed extensions to image morphing that correctly handle 3D projective camera and scene transformations. View morphing works by prewarping two input images, computing an image-morph (image warp and cross-dissolve) between the prewarped images, and then postwarping each in-between image produced by the morph. The prewarping step corresponds to changing the orientations of the two input views, but not the camera positions. The images are projected onto a common image plane that is parallel to the line between the two cameras' optical centers. From this special parallel camera configuration, we have shown that image morphing (i.e., linear image interpolation) produces physically-correct perspective views. These views correspond to positioning a virtual camera on the line between the original cameras, and orienting the camera parallel to the prewarped views. The postwarping step warps the morphed image so as to change the orientation of the virtual camera. This sequence of steps is illustrated in Figure 1.

The major contributions to view synthesis that are achieved by view morphing include:

- Ability to synthesize image sequences corresponding to linear camera motion between two basis images' known or unknown camera positions (with the orientation of

the camera along that path specifiable by the user)

- Can compute smooth transitions between any two images, regardless of source or content, producing simultaneous transitions in viewpoint, shape and color; consequently, the approach can perform both rigid and non-rigid transformations, and use a variety of types of input from photographs to drawings
- Does not require knowledge of 3D shape nor does it need calibrated cameras
- When a generic visibility assumption holds, which we call monotonicity, view morphing guarantees a unique, physically-correct solution for all viewpoints on the line between the optical centers of two input cameras
- When visibility changes occur between basis views, the monotonicity assumption is violated, but image quality degrades only locally and can be minimized by using different feature correspondences
- When a stronger version of monotonicity holds for a set of basis views, new views can be synthesized for all viewpoints within the convex hull of the input cameras' optical centers
- Efficient implementation of the algorithm is possible because many steps are 1D scanline operations

2.1 Evaluation Plan

Research results related to view morphing will be demonstrated and evaluated by both theoretical analysis and experimental testing of prototype systems. With respect to theoretical properties of interest, view morphing is currently limited in that it cannot directly cope with significant changes in visibility, is difficult to use for synthesizing a large range of views of a scene from many basis images, and can require solving the correspondence problem for views that are far apart. These issues will be investigated further. In addition, we intend to continue the development of our view morphing system implementation in order to decrease the processing time, require less user interaction, and improve the realism of the synthesized views. Experimental evaluation using VSAM-related data sets will be performed.

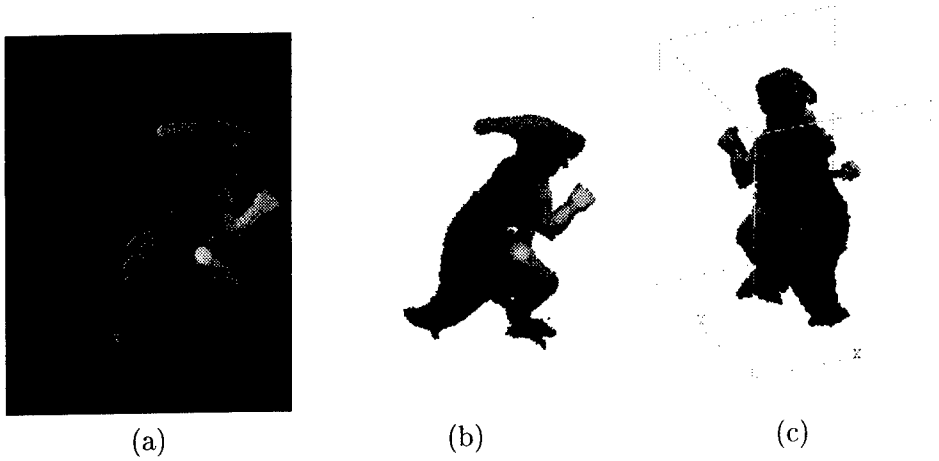
3 View Synthesis from Many Views

To synthesize new views from arbitrary camera viewpoints given a set of basis images is a difficult unsolved problem. One important requirement is the ability to integrate information from images containing significant differences in the parts of the scene that are visible. Second, since the desired results are photorealistic new views, methods must be "dense" so as to render images containing accurate texture and color information at every pixel, not a sparse set of feature descriptions. A third requirement is scalability—the capability for combining an arbitrary number of basis views, with corresponding improvements in the quality of the synthesized views.

With these requirements in mind, we are developing a new approach, called **voxel coloring** [Seitz and Dyer, 1997a, Seitz and Dyer, 1997b], that reconstructs the "color" (radiance) at points in an unknown scene. In our initial study we assume a static scene containing Lambertian surfaces under fixed illumination so the radiance from a scene point can be described simply by a scalar value, which we call *color*.

Coping with images with large changes in visibility means we must solve a difficult correspondence problem between images that are very different in appearance. Rather than use traditional approaches such as stereo, we use a scene-based approach. That is, we discretize 3D scene space into a set of voxels that are traversed and colored in a special order. The advantage of this is that simple voxel projection determines corresponding image pixels. The main disadvantage is that this requires precise camera calibration to achieve the necessary accuracy.

We have shown that certain voxels have an invariant color, constant across all possible interpretations of the scene that are consistent with the basis images. This leads to a volu-



(a) (b) (c)

Figure 2: Reconstruction of a dinosaur toy. (a) One of 21 original images taken from slightly above the toy while it was rotated 360°. (b-c) Two views rendered from the reconstruction.

metric voxel coloring algorithm that labels the color-invariant scene voxels based on their projected correlation with the input images. Correlation consistency will work only if we can determine when a voxel in fact corresponds to the projected pixel in an image, or whether the pixel corresponds to a different (closer) scene point, which occludes the current voxel. To solve this visibility problem, we introduce a geometric constraint on the input camera positions that enables a single visibility ordering of the voxels to hold for every input viewpoint. This is a relatively weak constraint in that it allows significant freedom in the placement of the input cameras, but it enables the visibility problem to be solved by simply traversing and labeling the voxels in increasing distance from the input cameras. Furthermore, the method is independent of scene complexity.

Putting this all together, the voxel coloring algorithm works as follows. The scene is initialized to a volume of voxels. These voxels are traversed layer-by-layer, where a layer contains all voxels that are equidistant from the cameras' convex hull. The layer closest to the cameras is visited first, and so on until the layer of voxels that is farthest from the cameras is considered. A voxel is processed by projecting it into each basis image and determining how well its corresponding image pixels' colors are correlated. If the correlation is above a threshold, the voxel is added to the reconstructed shape and labeled with the color of its pixels.

The final result is a dense, volumetric reconstruction, with associated color information, of scene surface points that is guaranteed to be consistent with all the basis images, regardless of visibility changes and scene concavities. Using this reconstruction, the scene can be rendered from any view by projecting the voxels in the desired direction. Figure 2 shows two views rendered using the reconstruction produced by a set of 21 input images.

The major contributions to view synthesis that are achieved by voxel coloring include:

- Reconstructs a dense description of scene surface points and associated color (radiance) values, which can be used for rendering arbitrary new views
- Uses color invariance to ensure that all voxels reconstructed are consistent with all of the basis images
- Allows input views containing large visibility differences by operating in scene (voxel) space and using a weak camera position constraint
- Permits widely separated input camera positions
- Scales up directly to an arbitrary number of basis views, with processing time linear in the number of input images

The reconstruction produced by voxel coloring can also be used for virtually *modifying* the reconstructed scene by image editing operations such as image painting, scissoring, and morphing. We call this **plenoptic image editing** [Seitz and Kutulakos, 1997] because the user can edit any one image and those changes are propagated automatically, in a physically-consistent way, to all other images as if the 3D environment had itself been modified. This allows a user to visualize how edits to an object via one image will affect the object's appearance from other viewpoints. The key component in realizing these operations is the reconstruction produced by voxel coloring. While preliminary results are encouraging, there are many open research problems that need to be addressed to make this approach more effective.

3.1 Evaluation Plan

Research results related to voxel coloring will be demonstrated and evaluated by both theoretical analysis and experimental testing of prototype systems. Improved methods are needed for handling large numbers of images, for example from a video stream, which are uncalibrated. Extensions are also needed to handle non-Lambertian scenes and dynamic scenes. The plenoptic image editing framework needs to be explored further to determine the types of scene modification operations that would be useful for VSAM applications. We plan to continue developing our system implementation so that the photorealistic quality, processing speed, scalability to large numbers of basis views, and large range of feasible output views are experimentally demonstrable. Also, can the method produce smooth and natural scene visualizations corresponding to a moving camera? Synthesized view quality assessment will be performed if data sets with ground truth are available.

References

[Seitz and Dyer, 1995] S. M. Seitz and C. R. Dyer. Physically-valid view synthesis by image interpolation. In *Proc. IEEE Workshop on Representations of Visual Scenes*, pages 18–25, 1995.

[Seitz and Dyer, 1996a] S. M. Seitz and C. R. Dyer. Toward image-based scene representation using view morphing. In *Proc. 13th Int. Conf. on Pattern Recognition, Vol. I*, pages 84–89, 1996.

[Seitz and Dyer, 1996b] S. M. Seitz and C. R. Dyer. View morphing. In *Proc. SIGGRAPH 96*, pages 21–30, 1996.

[Seitz and Dyer, 1997a] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proc. Computer Vision and Pattern Recognition Conf.*, 1997. To appear.

[Seitz and Dyer, 1997b] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. In these Proceedings.

[Seitz and Dyer, 1997c] S. M. Seitz and C. R. Dyer. View Morphing: Uniquely predicting scene appearance from basis images. In these Proceedings.

[Seitz and Kutulakos, 1997] S. M. Seitz and K. N. Kutulakos. Plenoptic image editing. Technical Report 647, Computer Science Department, University of Rochester, Rochester, NY, January 1997.

[Seitz, 1997] S. M. Seitz. Bringing photographs to life with view morphing. In *Proc. Imagina 97*, pages 153–158, 1997.

Image Understanding Research at Rochester

Christopher Brown, Randal Nelson
Computer Science Department
University of Rochester
Rochester, NY 14627-0226
brown@cs.rochester.edu, nelson@cs.rochester.edu

Abstract

Image Understanding research in pose estimation, object recognition, various forms of learning, predictive techniques, image-based view synthesis is continuing. The video surveillance and monitoring project we are just starting will tie together several of these techniques but requires new work.

The **objective** of the most recent IU research at Rochester is to enhance battlefield awareness through the augmentation of surveillance and monitoring capabilities by the automatic recognition of objects and activities, and then by presenting graphic presentations of them in live video. Several **research questions** arise: the familiar object- and activity-recognition problems; the problem of real-time tracking of natural features and the determination of the state (pose and velocity) of remote objects; the representation of objects for computer processing; the registration of graphic objects with live video; the role of prediction in robust recognition schemes. We **evaluate** our work by comparision with other competing schemes, by quantifying how each modification or technique affects system performance, and by quantifying the reliability, speed, effectiveness, and versatility of demonstration integrated systems.

For more information:
http://www.cs.rochester.edu/
http://www.cs.rochester.edu/darpa/main.html

1 3-D Object Recognition with Generalization

Appearance-based object recognition methods have recently demonstrated good performance on a variety of problems. However, many of these methods either require good whole-object segmentation, which severely limits their performance in the presence of clutter, occlusion, or background changes; or utilize simple conjunctions of low-level features, which causes crosstalk problems as the number of objects is increased. We have been developing an appearance-based method that utilizes intermediate-level features, in this case automatically extracted 2-D boundary fragments, to provide normalized keys and descriptors which alleviate these problems. The work is described more fully elsewhere in this Proceedings.

We have done various large-scale performance tests, involving, altogether, over 2000 separate test images ranging from sports cars and fighter planes to snakes and lizards over full spherical or hemispherical ranges (and planar scale, translation and rotation). In one experiment, we investigate performance scaling with increasing number of objects, and observe a decline in recognition accuracy from 99% to 97% as the number of objects increases from 6 (11 hemispheres) to 24 (34 hemispheres). In a second experiment, we investigate the effect of clutter on the performance of the recognition system. The third was a generic recognition experiment, where the system is trained on several objects in each of several several classes, (e.g. planes, snakes, cars), and asked to classify example objects from the same generic classes, but not in the training set. We get about 93% accuracy for 5 classes.

Appearance-based methods are a useful technique; however because matches are generally made to representations of complete objects, these methods tend to be sensitive to clutter and occlusion and require good global segmentation for success. Hough transform methods (and other voting techniques) allow evidence from disconnected parts to be effectively combined, but the size of the voting space increases exponentially with the number of degrees of visual freedom.

Our idea is to represent the visual appearance of an object as a structured combination of a number of semi-local features, or fragments. Under different conditions (e.g. lighting, background, changes in orientation etc.) the feature extraction process will find some of these, but in general not all of them. However, we show that the fraction that is found by feature extration processes is frequently sufficient to identify objects in the scene. This is how we cope with the notorious segmentation problem.

Our semi-invariant local objects are called *keys*. A key is any robustly extractable part or feature that has sufficient information content to specify a configuration of an associated object plus enough additional parameters to provide efficient indexing and meaningful verification. For rigid objects, configuration generally implies location and orientation, but more general interpretations can be used for other object types. Semi-invariant means that over all configurations in which the object of interest will be encountered, a matchable form of the

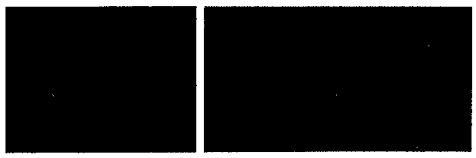


Figure 1: The type of object used in the scaling test set

feature will be present a significant proportion of the time. We currently make use of a single key feature type consisting of curve orientation templates normalized by robust boundary fragments. We call these features *curve patches.*

We use a hashed database (an associative memory) organized so that access via a key feature evokes associated hypotheses for the identity and configuration of all objects that could have produced it. These hypothesis are fed into a second stage associative memory, keyed by the configuration, which maintains a probabilistic estimate of the likelihood of each hypothesis based on statistics about the occurrence of the keys in the primary database. The idea is similar to a multi-dimensional Hough transform without the space problems.

The basic recognition procedure consists of four steps. First, potential key features are extracted from the image using low and intermediate level visual routines. In the second step, these keys are used to access the associative memory and retrieve information about what objects could have produced them, and in what relative configuration. The third step uses this information, in conjunction with geometric parameters factored out of the key features such as position, orientation, and scale, to produce hypotheses about the identity and configuration of potential objects. Finally, these hypotheses are themselves used as keys into a second stage associative memory, which is used to accumulate evidence for the various hypotheses.

The system can be trained efficiently from image data (Fig. 1). The process is efficient, and essentially runs in time proportional to the number of pairs stored in memory. Speedups are possible since the algorithm and database are readily parallelized [Hunt and Nelson 1996].

The notion of generic visual classes is ill defined scientifically. What we have is human subjective impressions that certain objects look alike, and belong in the same group (e.g. airplanes, sports cars, spiders, teapots etc.) Unfortunately, human visual classes tend to be confounded with functional classes, and biased by experience and other factors to an extent that makes formalizing such classes, even phenomenologically, very difficult. On the other hand, the subjective intuition is so strong, the early evidence of correct "generalization" so intriguing, and the payoffs so great, that we are looking into the matter seriously. The recognition system was trained on a subset of each class, and tested on the remaining elements. The training sets consisted of 4 cups, 2 airplanes, 2 jet fighters, 4 sports cars, and 4 snakes. We would have liked to have more samples of the planes, but local toy stores had no diversity. The performance is best for the cups, planes, and sports cars, all at around 95%. The fighter planes were the worst, the reason being that there is quite a bit of difference between the exemplars in some views in terms of armament carried, which tends to break up some of the lines in a way the current boundary finder does not handle. One of the test cases also has a camouflage pattern on it.

Future plans include adding enough additional objects to push the performance below 75%, both to observe the functional form of the error dependence on scale, and to provide a basis for substantial improvement. We also plan to complete an evaluation of performance in the presence of clutter of various forms. Finally, we want to experiment with adapting the system to allow fine discrimination of similar objects (same generic class) using directed processing driven by the generic classification.

2 Image Based View Synthesis

We have devised and demonstrated an image-based method that can produce images of arbitrary animated motions of an articulated agent from a model learned by just watching the agent performing some other unrelated task. No camera calibration, kinematic or 3D CAD model is needed. Instead the visual-model estimation (above) is combined with recent appearance-based visual representations. The basic idea (fig. 2) is to parameterize the appearance changes in motor space (e.g. joint angles of a robot or human) instead of a visual space (e.g. affine or projective). Appearance changes are related to motor space in a two stage process. First the changes in the input intensity image are converted into a compressed representation in an *appearance space* using either a "Nayar"-type linear subspace or a motion-based method. Second the visual-motor model between the changes in appearance space and motor space is learned. To generate movies the process is reversed, and a user-supplied motor program is converted to appearance space and then to images. The method can be used both on-line, augmenting real-time live video with synthesized images based on an instantaneous, on-line estimated model, or off-line, synthesizing movies from a previously learned model.

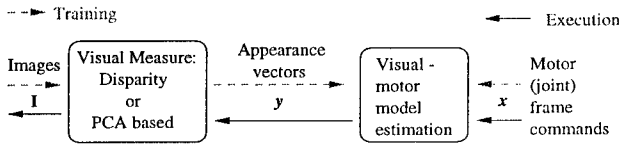


Figure 2: Outline of our method.

In current work we have been able successfully to estimate the visual motor models for up to three joint articulated robot and human arms (Fig. 3, and a simulated movie at

`http://www.cs.rochester.edu/u/jag/SimAct/SimAct.html}`

and

`{\tt ../handmovie1_1_1.mpg}`

), and use these models to simulate arbitrary motions. Future work includes using visual representations based

on modern image sequence compression techniques, and trying to extend the method to more complex agents.

Figure 3: Simulation of a human arm.

3 Video-Based 3D Scene Capture and Editing

An increasing number of applications, from battlefield awareness to scene visualization and interior design, require building and interacting with photorealistic models of physical 3D objects and environments. The availability of low-cost, high-performance rendering and video acquisition hardware promises a new generation of multimedia tools and techniques for this purpose that use photographs or video images as input, exploit their ability to interact with the user or the user's environment, and are robust and cost-effective. To this end, we are investigating novel scene representation and reconstruction methods whose goal is to make the digital capture and manipulation of physical 3D scenes as easy and efficient as the task of digitizing and editing a single color photograph with Adobe Photoshop or a similar picture editor. Research at the University of Rochester on this topic is outlined below.

3.1 Scene Capture and Editing by Plenoptic Decomposition

We have developed a new approach for capturing and editing 3D objects and environments from a few color images taken at known camera positions [Seitz & Kutulakos 1997]. its basic form, the approach uses the input images to automatically build a representation from which photorealistic views of a scene can be synthesized for *any* viewpoint. Our work overcomes a number of limitations of existing techniques, including the need to manually establish correspondences between images of complicated scenes, the inability to generate physically-correct views of the scene for viewpoints away from the input images, and the need to acquire a large set of images that cover all viewpoints. A unique feature of the approach is that, in addition to providing this basic functionality, it allows editing a captured scene to achieve various 3D effects (e.g., animating 3D objects in a scene, changing their color, or removing them entirely). In order to achieve this, we have introduced a new class of image editing operations that extends operations such as image painting, scissoring, and morphing found in common picture editors. The operations are specifically designed to maintain consistency across all views of a physical 3D scene; unlike their counterparts in programs such as Photoshop, edits to any one image propagate automatically to all other views as if the (unknown) 3D scene had itself been modified (Fig. 4).

At the heart of the approach lies a novel volumetric technique called *plenoptic decomposition* that enables an object's plenoptic function to be reconstructed from an incomplete set of camera viewpoints. Plenoptic decomposition attempts to interpret the rays of light hitting the input images in terms of a 3D shape and an unknown but slowly-varying reflectance function. This is accomplished by volume carving: the unknown scene is initially represented by a trivial, voxelized shape (e.g., a cube), and voxels that do not conform to an *a priori*-specified reflectance model are iteratively carved away. Our main results are that (1) the resulting shape-reflectance representation is sufficient for synthesizing arbitrary, photorealistic views of a scene, (2) the representation provides sufficient point correspondence information to consistently propagate 2D image edits to all views of a 3D scene, and (3) the recovery of a shape- reflectance representation from two or more images is always a solvable and well-posed problem. In its current implementation, our system allows physical 3D objects to be captured by rotating them in front of a stationary camera, it allows the synthesis of arbitrary views even when only two images are given as input, and it supports interactive operations such as image painting and light source modification.

3.2 Global Surface Reconstruction from Occluding Contours

It is well-known that the occluding contour is a rich source of information about an object's 3D shape. We have recently shown that the problem of recovering global shape from occluding contours becomes considerably simplified if it is formulated as a reconstruction problem in the space of oriented rays that intersect the object [Kutulakos 1997]. The approach works by first mapping every pixel on every occluding contour image of a rotating object to a point in *ray space* (the oriented projective sphere $\mathcal{T}^2$). Using results from oriented projective geometry and the theory of convex duals, we have shown that if all occluding contour pixels are mapped to a set of points in ray space, global surface reconstruction can be achieved by applying well-known and efficient convex hull and line arrangement algorithms to that point set. The approach works with both calibrated and uncalibrated cameras, does not assume the ability to establish correspondences between occluding contour curves across frames, does not impose an *a priori* voxelization of 3D space, and can enforce global shape constraints such as convexity. Topics currently under investigation include the use of robust convex hull techniques to enhance the method's insensitivity to noise and out-

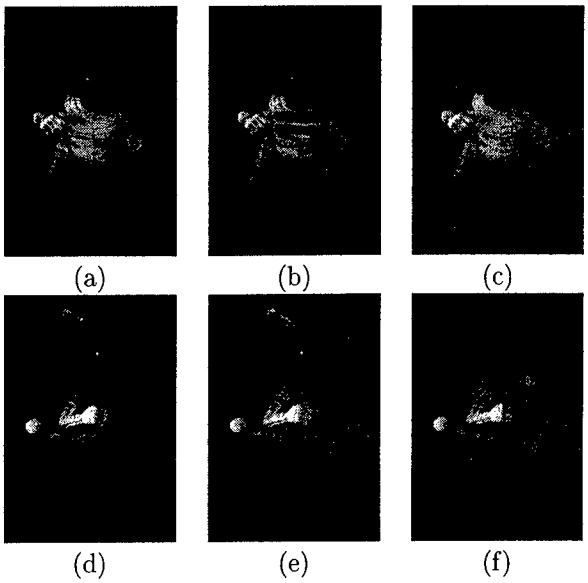


Figure 4: Scene editing examples. (b),(c) show image painting and scissoring operations, respectively, applied to image (a) of a dinosaur toy. (e),(f) show the images that were automatically generated by propagating the operations to image (d). Observe that the propagation properly accounts for difference in visibility between the two views (e.g., part of the painted area is correctly occluded by the dinosaur's right hand in image).

liers, and the integration of the method with techniques such as plenoptic decomposition that can refine a computed shape by reconstructing surface regions that never project to the contour.

4 Learning

Learning in all its forms continues to be a central aspect of our research. Only a small proportion of this work is directly relevant to the current image understanding effort. Some of the most relevant is learning applied to manipulation, vehicle control, and navigation. We are using high degree of freedom systems (up to 22 dof) in complex manipulation tasks. Several learning and adaptive techniques are deployed, including genetic algorithms, Jacobian updating, neural net techniques for function approximation, and novel sparse distributed memory implementations [Fuentes and Nelson 1996a,b; Jagersand et al 1996, 1997; Jagersand and Nelson 1996; Rao and Fuentes 1996].

Learning is also a key component of our basic vision research and has played a role in translating results from visual homing to object recognition as well [Nelson 1996a,b].

5 Augmented Reality and Surveillance

Work is just beginning that is aimed at increasing the effectiveness of video surveillance and monitoring (VSAM) exploitation by the new visualization technique of view augmentation, which has produced dramatic improve-

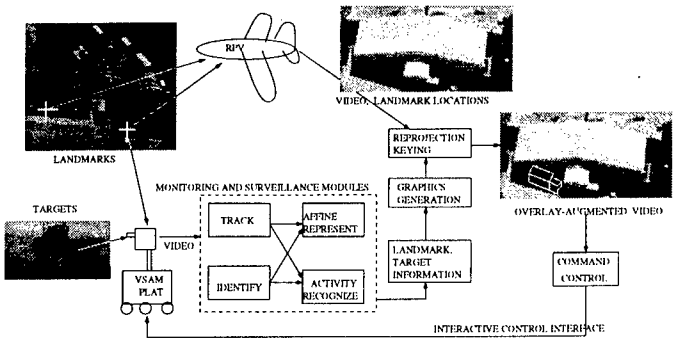


Figure 5: Graphic overlays depicting targets and utilities augment live video from an RPV. An observer-commander has an interactive interface for VSAM observation and command. Video analysis can take place on the VSAM, at the center, or at both sites.

ments in operator performance in other domains (e.g. medical). In an augmented view, the operator or field personnel see a graphic rendition of the target correctly registered with a live video stream or a canonical view, even if the target is obscured or hidden in the video (Fig. 5). This calibration-free view augmentation relies on new work in non-Euclidean object representations and on accurate feature tracking [Kutulakos and Vallio 1996a,b]. Features of landmarks are tracked to establish the non-Euclidean frame, and target features are tracked to locate the target in the frame. The context-based tracker monitors the configuration of targets, the scene context, and the performance of trackers to choose the best technique for the current situation [Brown 1996b].

Currently, camera calibration is considered necessary for augmenting video streams with correctly-registered view-dependent graphical information. In the mobile VSAM scenario, calibration objects may not be visible, and platform movements and multiple sensor degrees of freedom make open-loop calibration impractical. View augmentation for scenes at different ranges or resolutions is beyond the current state of the art. To achieve scalable, calibration-free view augmentation we shall exploit three new ideas: (1) representing the internal reference frames of the cameras and the graphical models within a *non-Euclidean frame of reference (affine or projective)*, (2) establishing a reference frame by *tracking detectable features* in the live video stream whose 3D configuration in space is unknown, and (3) *using a hierarchy of non-Euclidean reference frames* for representing the entire VSAM environment, depending on the geometry and dimensions of the environment visible from any given camera [Kutulakos and Vallino 1996a,b].

Our challenge is to move view augmentation to uncalibrated environments, and we plan to use non-Euclidean (affine and projective) representations that render calibration unnecessary. Instead, tracking of landmarks is substituted [Araujo and Brown 1996; Araujo et al. 1996].

We plan to extend our work on repetitive activity recognition [Polana and Nelson to appear] to encompass new activity types including impulsive activities like throwing an object, and transitional activities like start-

ing or stopping a vehicle. We want to use situational context to constrain classifications and decrease type I and type II errors.

The same graphics that augment the view provide a means for annotation of targets or regions of 3-D space and for interactivity; they can be used to build a unified observation and command interface for more effective deployment, positioning, cueing, and control of the semi-autonomous VSAMs.

Mobile VSAM platforms are deployed in an area; perhaps a remotely piloted vehicle (RPV) platform circles above. Targets and their activities are sometimes visible, sometimes invisible to any given VSAM or the RPV. At a remote observation and command center an *augmented view* is presented featuring graphic overlays on the live video stream from a VSAM, RPV, or other observation post. In the augmented view the targets, activities, or other features of interest appear visible "through walls", "through hills", or "through cover". The graphic overlays are correctly registered with the live video without open-loop platform calibration techniques or visible calibration objects (Fig. 5).

A summary of the approach is the following.

1. Use outdoor-domain trackers and outdoor cameras to apply the technology to buildings, vehicles, and campus activities.
2. Combine affine and projective view algorithms to deal with a wide variety of imaging situations (ranges, focal lengths, resolutions).
3. Input will be video streams from mobile platforms in outdoor situations.
4. Deal with moving targets, moving VSAMs, and moving frames of reference.
5. Use interactive graphics techniques to provide observer annotation and command interface.
6. Provide graphic overlays not just for objects but for activities.
7. Improve graphics with better models and advanced capabilities like obscuration and shading.

Augmented views can serve not just to inform an observer but to provide a command interface to VSAMs. Our idea is an integrated system of multiple VSAMs and other observation platforms along with an interactive command center. The observer-commander sees augmented views, and communicates **what** the VSAM should do. The VSAM, using local sensing and control, takes care of **how**. Such semi-autonomy has many advantages over pure teleoperation (for which problems arise with bandwidth, remote sensing, and delays) or pure autonomy (which is still a research area) [Ballard et al. 1996]. We are looking into advanced haptic feedback as well as graphical techniques (Section 9).

Activity recognition is an important component of this project for three reasons: (1) It serves to focus attention, to alert the VSAM to important phenomena and to allow it to ignore unimportant ones (such as wind-blown foliage), (2) activities are of interest to remote commanders and observers, and communicating their type and properties is more efficient than sending live images and (3) it can be used for local decision-making, allowing VSAMs to cooperate with each other and to make decisions based on sensed activities. Past work [Polana and Nelson to appear] has enabled us to recognize "natural clutter", or temporal textures (wind-blown foliage, fire, water), and repetitive activities like walking.

We plan to demonstrate the VSAM capabilities on our two indoor-outdoor mobile platforms (Section 9).

6 Pose Estimation

The key idea of a classic pose recovery algorithm by Lowe is that, given a certain estimate for the unknown pose parameters ($\mathbf{x}$) — and possibly even for some internal degrees of freedom in the scene — one can reproject the known 3D scene model into the image plane and then compute a vector of errors ($\mathbf{e}$) between the positions, orientations and / or apparent angles of the reprojected features, and the corresponding actual measurements. Then, the partial derivatives of those image measurements with respect to the free pose and scene parameters are approximated locally by a jacobian (first order) matrix $\mathbf{J}$. This allows one to compute a vector of corrections needed to adjust $\mathbf{x}$ to the image measurements in a least–squares sense, according to the local linearization implied by $\mathbf{J}$:

$$\mathbf{J}\,\delta\mathbf{x} = \mathbf{e}, \qquad \text{where: } \mathbf{J}_{ij} = \frac{\partial e_i}{\partial x_j}. \tag{1}$$

By replacing $\mathbf{x}$ with $\mathbf{x} + \delta\mathbf{x}$ and iterating the procedure, one can then obtain a set of values for the free parameters ($\mathbf{x}$) that minimize the euclidean norm of the vector of reprojection errors ($\mathbf{e}$) locally, in the parameter space.

The main weakness of Lowe's original formulation is the geometrical model suggested for computation of the jacobian matrix $\mathbf{J}$. Lowe suggested that, in order to achieve greater efficiency, one should reparametrize the translational components of the pose, so as to express them directly in image plane coordinates, rather than in a tridimensional euclidean space, as usual. More specifically, in his model, the image coordinates of an arbitrary feature (u and v) are expressed as a function of the corresponding model–space coordinates ($\mathbf{p}$) by the following equation:

$$\begin{aligned} (x', y', z')^T &= \mathbf{R}\,\mathbf{p}, \\ (u, v) &= \left(\frac{fx'}{z'+D_z} + D_x,\ \frac{fy'}{z'+D_z} + D_y\right), \end{aligned} \tag{2}$$

where D_x, D_y and D_z are the redefined translational parameters and $\mathbf{R}$ is a 3×3 orthonormal rotation matrix, as usual. So, Lowe's formulation assumes that D_x and D_y are constants to be determined by the iterative procedure, when in fact they are not constants at all — they depend on the depth of each individual feature with respect to the camera reference frame.

We drop this restriction [Araujo and Brown 1996, Araujo et al 1996] by redefining:

$$\begin{aligned} (x', y', z')^T &= \mathbf{R}\,\mathbf{p}, \\ (u, v) &= \left(f\frac{x'+t_x}{z'+t_z},\ f\frac{y'+t_y}{z'+t_z}\right), \end{aligned} \tag{3}$$

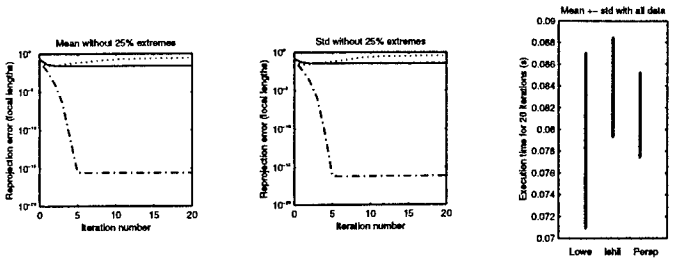


Figure 6: Left and center: convergence of an image–space error metric, the Norm of Distances Error, with respect to the number of iterations of Lowe's (solid line), Ishii's (dotted line), and our full projective solution (dash–dotted line); statistics exclude the best and worst 25% results. Right: mean and standard deviation of execution time for 20 iterations of each method; statistics include all data.

where the translation vector $\mathbf{t} = (t_x, t_y, t_z)^T$ is now measured in a 3D euclidean space.

A careful empirical comparison involving several models described in the literature, as well as a reformulation of Lowe's algorithm proposed by Ishii, showed that that the use of true perspective yields a much more accurate numerical technique, with superexponential convergence for a wide range of initial conditions and even arguably better computation–time properties, as illustrated by Fig. 6.

We are also working on decoupling rotation and translation recovery with point correspondences. Christy and Horaud proposed two modifications to Lowe's algorithm: **(i)** the use of quaternions, rather than RPY or Euler angles to represent orientation, in order to eliminate cetain singularities; and **(ii)** the use of second order terms in the local approximation to the error surface, in order to improve the global convergence properties of the resulting algorithm. As a side effect of those modifications, they are able to decouple the recovery of the rotational components of the pose from the recovery of the translation, so that the original 6D problem can be reduced to two much simpler 3D problems.

However, their decoupled solution is based on edge correspondences and can not effectively explore additional constraints provided by actual point correspondences. This results in only a slight loss of accuracy in the general case, but in some extreme scenaria (quasi–planar objects) it can result in catastrophic failures, due to the presence of a singularity in the pose recovery process. We showed that the use of additional constraints provided by point correspondences allows one to eliminate that singularity, while still keeping the rotation recovery decoupled from translation, for improved efficiency.

7 Hierarchical Prediction Improves Robustness in Several Applications

Computational theories of biological mechanisms are under active investigation in Dana Ballard's laboratory [Rao 1996; Rao and Ballard 1996a,b,c,d,e; Rao et al. 1996; Zelinsky et al. 1996]. Much of this work has practical significance in producing more robust and powerful computer vision, salience, and recognition algorithms. One focus of current work is a model of the cortex as a hierarchical predictor that provides more robust, general, and adaptive results by dynamically predicting the input. The reciprocity of connections between cortical areas brings to mind the sort of feedback of observations to predictions that one sees in a Kalman filter (or in the Baum-Welch algorithm). The Kalman filter analogy is a powerful one that can be translated into mathematics, algorithms, and biological mechanisms. For instance, continuous adaptation of synaptic strength by a hierarchical Kalman filter minimizing prediction errors provides a description that generalizes several proposed neural encoding schemes and provides functional interpretations for several well-known psychophysical and neurophysiological phenomena [Rao and Ballard 1996e].

The Kalman filter approach can be adapted to produce a generative mechanism that allows rotation-, scale-, and translation-invariant recognition as well as application to stereopsis and motion estimation [Rao and Ballard 1996a]. The Kalman filter model is derived from MDL considerations. When combined with a Hebbian model of adapting synaptic weights using a learning rule also derived from the MDL principle, can explain experimental neurophysiological observations in free viewing and fixating conditions [Rao and Ballard 1996d].

The Kalman filters can be made robust to structured and unstructured noise by allowing the measurement covariance matrix to be a nonlinear function of the prediction errors. Simultaneously the filter is learning an internal model of input dynamics by adapting both measurement and state transition matrices using two adaptation rules. This treatment generalizes and renders more responsive the traditional "gating" techniques, and is shown to work for segmentation and recognition of objects and also of image sequences in noise, and with clutter and occlusion [Rao 1996].

8 Image Understanding Environment

We have installed version 1.3.1. of the IUE along with some of the edge-finding sample application programs, and have designed a Hough Transform class that uses the Histogram class. Our initial white paper [Brown 1996a] was wide of the mark in several places, and it is being rewritten to accord with our latest thinking and the working implementation. The HT class takes image evidence (currently and most commonly this is a collection of IUE_edgels) and produces an IUE_histogram that records vote strengths in parameter space and a similarly-dimensioned and similarly-sized IUE_accumulator_array whose elements contain the set of evidence items that voted for a particular parameter space bin. Voting patterns are determined by a function that is passed in at HT object creation time. We have sample voting functions for lines and circles.

The HT application calls for several extensions to current IUE classes or implementations. Among the most obvious are N-dimensional arrays (for N-dimensional parameter spaces) and histograms with floating point (instead of integer) elements (for recording floating point

vote strengths). Quite useful will be histogram smoothing and peak-finding functionality that is specced for inclusion in the class. In future work we want to produce some more sophisticated voting functions (for example, to use edgel properties such as uncertainty in location or direction).

9 Laboratory Developments

The past year has seen a qualitative enhancement in the laboratory facilities with the arrival of a "memory machine" configured to support Nelson's object recognition work. It is a four-processor ultraSparc with 2Gb of main memory. Another major acquisition (using DARPA DURIP funds), was two automated wheelchairs manufactured by KIPR, each equipped with a wireless ethernet, FM television broadcasting, twin-Pentium Linux system, infrared, sonar, odometric and video sensors. These platforms are to support VSAM research. We have four "Phantom" haptic VR devices, which sample 3-D position of a human finger and apply force back to it at 1500 Hz. They thus allow the simulation of realistic interaction with table-top objects whose dynamics, texture, and other "feelable" properties are under computer control. We hope to use the Phantoms as part of a human-computer interface for control of remote manipulations or to command vehicle movements.

10 Acknowledgements

This work was supported by DARPA contract MDA972-92-J-1012, DARPA contract DAAL03-91-C-0034 Task 95228, NSF Postdoctoral Fellowship CDA 9503996, and NSF IIP hardware grant CDA-94-01142.

References

Araújo, H. and C.M. Brown, "A note on Lowe's tracking algorithm," TR 610 (replaced by TR 641), Computer Science Dept., U. Rochester, April 1996.

Araújo, H., R.L. Carceroni, and C.M. Brown, "A fully projective formulation for Lowe's tracking algorithm," TR 641 (replaces TR 610), Computer Science Dept., U. Rochester, November 1996.

Ballard, D.H., M.M. Hayhoe, P.K. Pook, and R.P.N. Rao, "Deictic codes for the embodiment of cognition," NRL TR 95.1, Nat'l. Resource Lab. for the Study of Brain and Behavior, U. Rochester, revised July 1996; to appear, *Behavioral and Brain Sciences*.

Brown, C.M., "The Hough transform and the IUE," IUE-TC Working Paper, December 1996a.

Brown, C.M., "Modern computer vision and augmented reality," invited talk, *ICPR*, Vienna, Austria, September 1996b.

Fuentes, O. and R.C. Nelson, "Learning dextrous manipulation skills for multifingered robot hands," TR 613, Computer Science Dept., U. Rochester, revised October 1996a.

Fuentes, O. and R.C. Nelson, "Learning dextrous manipulation skills using multi-sensory information," *Proc., 1996 IEEE / SICE / RSJ Int'l. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, Washington, DC, December 1996b.

Hunt, G.C. and R.C. Nelson, "Lineal feature extraction by parallel stick growing," *3rd Int'l. Workshop on Parallel Algorithms for Irregularly Structured Problems* (IRREGULAR96), Santa Barbara, CA, August 1996.

Jägersand, M., O. Fuentes, and R.C. Nelson, "Acquiring visual-motor models for precision manipulation with robot hands," *Proc., 4th European Conf. on Computer Vision* (ECCV96), 603–612, Cambridge, England, April 1996.

Jägersand, M., O. Fuentes, and R.C. Nelson, "Experimental evaluation of uncalibrated visual servoing for precision manipulation," to appear, *Int'l. Conf. on Robotics and Automation*, Albuquerque, NM, April 1997.

Jägersand, M. and R.C. Nelson, "On-line estimation of visual-motor models using active vision," *Proc., ARPA Image Understanding Workshop*, Palm Springs, CA, February 1996.

Kutulakos, K.N., "Shape from the light field boundary," to appear, *Proc., Computer Vision and Pattern Recognition Conf.*, Puerto Rico, June 1997.

Kutulakos, K.N. and J.R. Vallino, "Affine object representations for calibration-free augmented reality," *ARPA Image Understanding Workshop*, Palm Springs, CA, February 1996; *Proc., IEEE Virtual Reality Annual Int'l. Symp.*, 25–36, Santa Clara, CA, April 1996a.

Kutulakos, K.N. and J.R. Vallino, "Non-Euclidean object representations for calibration-free video overlay," *Proc., Int'l. Workshop on Object Representation for Computer Vision*, Cambridge, England, April 1996b.

Nelson, R.C., "From visual homing to object recognition," in J. Aloimonos (Ed.). *Visual Navigation.* Lawrence Erlbaum, Inc., 1996a.

Nelson, R.C., "Visual learning and the development of intelligence," in S.K. Nayar and T. Poggio (Eds.). *Early Visual Learning.* Oxford, UK: Oxford U. Press, 215–236, 1996b.

Nelson, R.C. and C.M. Brown, "Real-time recognition and visual control: Image understanding research at Rochester," *Proc., ARPA Image Understanding Workshop*, Palm Springs, CA, February 1996.

Polana, R. and R.C. Nelson, "Detection and recognition of periodic, non-rigid motion," to appear, *Int'l. J. Computer Vision.*

Rao, R.P.N., "Robust Kalman filters for prediction, recognition, and learning," TR 645, Computer Science Dept., U. Rochester, December 1996; submitted for journal publication.

Rao, R.P.N. and D.H. Ballard, "A class of stochastic models for invariant recogntion, motion, and stereo," NRL TR 96.1, U.

Rochester, June 1996a; submitted for conference publication.

Rao, R.P.N. and D.H. Ballard, "A computational model of spatial representations that explains object-centered neglect in parietal patients," in J. Bower (Ed.). *Computational Neuroscience '96* (Cambridge, MA, July 1996). Plenum Press, 1996b.

Rao, R.P.N. and D.H. Ballard, "Cortico-cortical dynamics and learning during visual recognition: A computational model," in J. Bower (Ed.). *Computational Neuroscience '96* (Cambridge, MA, July 1996). Plenum Press, 1996c.

Rao, R.P.N. and D.H. Ballard, "Dynamic model of visual recognition predicts neural response properties in the visual cortex," NRL TR 96.2, Nat'l. Resource Lab. for the Study of Brain and Behavior, U. Rochester, August 1996d; *Neural Computation 9*, 4, 1997, in press.

Rao, R.P.N. and D.H. Ballard, "The visual cortex as a hierarchical predictor," NRL TR 96.4, Nat'l. Resource Lab. for the Study of Brain and Behavior, U. Rochester, September 1996e; submitted for journal publication.

Rao, R.P.N. and O. Fuentes, "Learning navigational behaviors using a predictive sparse distributed memory," *Proc., 4th Int'l. Conf. on Simulation of Adaptive Behavior: From Animals to Animats*, MIT Press, Cape Cod, September 1996.

Rao, R.P.N., G.J. Zelinsky, M.M. Hayhoe, and D.H. Ballard, "Modeling saccadic targeting in visual search," in D. Touretzky, M. Mozer, and M. Hasselmo (Eds.). *Advances in Neural Information Processing Systems 8* (Proc., NIPS 95, Denver, CO, November 1995). Cambridge, MA: MIT Press, 1996.

Seitz, S.M. and K.N. Kutulakos, "Plenoptic image editing," TR 647, Computer Science Dept., U. Rochester, January 1997.

Zelinsky, G.J., R.P.N. Rao, M.M. Hayhoe, and D.H. Ballard, "Eye movements during a realistic search task," abstract, *Invest. Ophthal. Vis. Sci. 37*, 1996.

Multi-Sensor Representation of Extended Scenes using Multi-View Geometry *

Shmuel Peleg Amnon Shashua
Daphna Weinshall Michael Werman
Institute of Computer Science
The Hebrew University of Jerusalem
91904 Jerusalem, Israel
{peleg,shashua,werman,daphna}@cs.huji.ac.il

Michal Irani
Dept. of Mathematics
Weizmann Institute of Science
Rehovot, Israel
irani@wisdom.weizmann.ac.il

Abstract

In order to provide visual surveillance and monitoring (VSAM) systems with tracking and visualization capabilities, we will develop algorithms that will address the problems and limitations of the existing state-of-the-art technology. In particular, we will:
(i) Develop image alignment techniques based on the *Trilinear Tensor*, both for single and for multiple sensors, to allow for accurate correspondence across sensors and time.
(ii) Develop alignment techniques of sequences obtained by sensors of different modalities.
(iii) Develop *Scene Manifolds* as extended scene representations that will efficiently combine information from single/multiple sensors. These representations will apply for a wide range of scenes and camera motions.
(iv) Develop algorithms for detection and continuous tracking of moving objects across time and sensors. The detection of moving objects will be performed using *Parallax Geometry* constraints.
(v) Develop means for the operator to visualize the scene. These include: *Scene manifold projection*, *novel view synthesis*, and *video stabilization*.

1 Introduction

The proposed approach is based on new techniques that have been pioneered and developed by the authors during the past three years, to represent and manage image information from multiple views. These techniques rely in part on the use of multilinear constraints across three or more views and the trilinear tensor [15, 19], parallax geometry [6], and the duality of camera and scene invariants [6, 21]. Applications to new view generation and to image stabilization are discussed in [13, 1]. Theoretical analysis shows that these methods are superior to existing two-frame methods, as they are free of degenerate scene or camera configurations [17]. A novel and efficient motion recovery method has also been developed [12]. Implementations of these methods demonstrated superior numerical stability for purposes of new view generation and of video stabilization. In addition, the "manifold projection" method [11, 14] can provide "scene manifolds", extended panoramic views of a scene without the usual distortions and loss of resolution associated with the traditional perspective image mosaics. These capabilities are detailed below.

1.1 Expected Impact

This research handles fundamental issues in multiple view correspondence, moving object detection, and visualization. As such, it will have a direct impact on VSAM technology and systems, as well as advancing Image Understanding technology in general. It has the potential of becoming the standard of all IU systems that deal with three-dimensional scenes acquired by two-dimensional sensors. In general IU contributions, correspondence techniques become more robust as geometry and photometry are combined together; view-synthesis methods use the 2D imagery directly without the need for a detailed 3D model [1]; video stabilization can be handled robustly under the most general situations (unlike existing methods) [13]; moving object detection from moving cameras can be handled using our duality of scene and motion invariants [6, 21, 7]; and extended scene representations will become more general than today's 2D mosaicing techniques [11] and more efficient than 3D CAD-based techniques. The use of true multi-view approaches will enable a simple and efficient way of relating infor-

*This research was funded by DARPA through the U.S. Office of Naval Research under grant N00014-93-1-1202. URL: http://www.cs.huji.ac.il/labs/vision

mation across frames and across sensor modalities.

For VSAM system, The proposed technology will provide some of the basic capabilities that are necessary to make the above scenario realizable. In particular, it will provide techniques for the following capabilities: (i) Construct a view-based scene layout from a single sensor. (ii) Accurately compute correspondences between images obtained by different sensors as well as by the same sensor at different times. This is necessary for the creation of a view-based scene representation and for coordination between the sensors. (iii) Combine information from multiple sensors over time into a single or a few *scene manifolds*, which provide an optimal extended view-based scene representation. Such a scene representation facilitates cooperated and continuous tracking of moving objects across sensors and time. It provides the operator with the means to visualize the scene and to provide scene-based instructions to the sensors on the VSAM platform. (iv) Detect moving objects in a wide range of scenes, while the sensor itself is moving. This provides the capability to detect potential enemy movement while the VSAM platform is on the move. (v) Synthesize new views from the constructed scene manifold for display to the operator. It will allow the operator to view the scene also from viewing positions where no sensor physically exists. (vi) We are presently also exploring a multi-frame method which involves the tracking of salient temporal changes in the image. This method proved effective in tracking many independently moving objects, like ants on the forest floor or people in a crowd [4].

2 Tensor-based Image Alignment

Three views give rise to a set of trilinear matching constraints that first became prominent in [15], whose coefficients form a tensor ("trilinear tensor") which encodes the relative camera locations, and whose underlying theory has been studied intensively. The tensor elements can be linearly recovered from at least 7 matching points across three views, and the tensorial equations (trilinearities) provide a matching constraint for use in image alignment tasks (such as "image transfer" applications).

In [20] we show that the tensor can combine the information of spatial-temporal derivatives instead of matching points with the geometric constraint of camera motion. The result is the introduction of a "Tensor Brightness Constraint" which is a parametric equation for solving for ego-motion directly from the spatio-temporal derivatives of the image sequence. In [2] we show that the fundamental matrix of two views is embedded in the tensorial equation as a rank-2 trivalent tensor, and we introduce a standard set of tensorial operators that apply to both two views and three views alike. In [17] we show that there are *no critical surfaces* for the computation of the trilinear tensor. The result of lack of degeneracy is powerful and suggests that three frames should be adopted as a "unit of analysis" rather than concatenation of pairs of views.

3 Multi-Sensor Alignment

In images acquired by sensors of *different modalities*, the relationship between the brightness values of corresponding pixels is complex and unknown. Contrast reversal may occur between the two images in some image regions, while not in others. Visual features present in one sensor image may not appear in the other image, and vice versa. Moreover, multiple brightness values in one image may map to a single brightness value in the other image, and vice versa. In other words, the two images are usually not *globally* correlated, i.e., they are not correlated in their entirety.

Image sequences obtained by sensors of different modalities have, however, additional common information components: (i) the scene geometry, and (ii) the scene dynamics. These two components are invariant to camera geometry, camera motion, and sensing modality, and therefore provide powerful constraints for multi-sensor image alignment. Yet, these have been rarely used for multi-sensor alignment. These two constraints will be added to our ongoing multi-sensor alignment work, which currently exploits only appearance information, and is reported in [8].

4 Parallax Geometry for Moving Object Detection

Moving object detection is a difficult problem in general. It has been shown to have robust solutions for specific cases. In particular, in cases where induces camera motion can be described parametrically (e.g., [9]), or in cases where the induced 3D scene information in the image plane is dominant relative to effects of moving objects. However, where the 3D information is comparable in size to that of the independent motion information, then the moving object detection problem becomes a very difficult one (e.g., Thompson et. al).

Parallax Geometry is a recently developed theory which provides a basis for 3D scene analysis (i.e., 3D shape recovery and moving object detection in 3D scenes), even in difficult scenarios where estimation of scene geometry or camera geometry is very difficult (e.g., when the 3D information in the image plane is sparse relative to effects of independent motions). It

provides a means for recovering 3D structure without the need to estimate camera motion (or camera geometry), and the means for detecting inconsistent 3D motions (i.e., independently moving objects) without the need to estimate neither camera motion (or camera geometry), nor scene geometry. For more details see [6].

The parallax geometry provides a multi-frame rigidity constraint which is *complimentary* to the epipolar rigidity constraint. Combining these two constraints together is expected to provide a more powerful rigidity constraint for moving object detection.

5 Scene Manifolds

"Scene Manifolds" is an alternative scene representation to image mosaics, which are currently used to represent panoramic view by stitching together a sequence of images [5]. Traditional image mosaics use a projection of all images into a common mosaic-image-plane, thus creating unacceptable distortions for images whose original image-plane is substantially different in angle from the mosaic-image-plane. In "Manifold Projection" the images are not projected into a plane, but are projected onto a *manifold* such that for every image its image-plane is tangential to the manifold. This manifold projection minimizes image distortion and provides superior visualization. Initial experiments with the manifold projections are very promising in terms of visualization quality as well as speed of computation.

The coordinate system and the surface on which the scene manifold is constructed depends mostly on the camera motion, but also on the scene structure. Some examples follow: (i) When a camera is rotating about a point behind the image plane, the optimal manifold for the projection is a ball centered at the rotation point. (ii) When a camera is translating in a plane, always looking perpendicular to the plane, the "manifold projection" will be an orthographic projection onto that plane. (iii) When the camera is zooming or is moving forward, the manifold is a cylinder [14].

6 Visualization

6.1 Video Stabilization

Another application of the "3 views as unit of analysis" is for video stabilization, which is a necessary component for building a visualization engine of the project. The technical problem behind video stabilization is the extraction and subsequent cancelation of the rotational component of camera motion across the video sequence. Most existing methods take one of the following two approaches. One approach is to compute the camera rotation only after computing the camera translation (the epipole) [19, 10]. The second approach assumes a specific 3D scene structure, e.g. assuming the existence and the detection of 3D planes in the scene [10, 18].

We have shown [13] that the camera rotation can be extracted directly from the tensor coefficients *without* the need to first recover the translational component of camera motion. We thus obtain a scheme that works under the most general conditions (without making assumptions on scene structure), and which does not suffer from the instability of recovering the translation from two views with a small base-line (see section on critical surfaces about instability of two views). This method was implemented and shown to provide robust video stabilization under general conditions [13].

6.2 Novel View Synthesis

Another application of interest is the possibility of generating views from novel viewing positions from a collection of other views. The two extreme ends of this task is on one end creating a detailed 3D map of the scene and with appropriate texture mapping and rendering to create a virtual "fly-through", and on the other end to use 2D mosaicing methods for stitching together images assuming only 2D transformations. The approach we have been taking is to synthesize views without creating a 3D model in the process, and relying instead on the implicit 3D information embodied in the trilinear relationships across three views. In this manner one could enjoy the simplicity of the 2D mosaicing techniques, yet be able to handle the general problem of synthesizing physical views from any viewing position. The technical idea is based on the fact that under certain arrangements of views, the trilinear tensors of triplets of views live in a small dimensional subspace [16]. One implication of this result is that one can *synthesize* new tensors from a given tensor and user specification of the position of a novel camera. Further details and demonstrations can be found in [1].

7 Local Curve Matching and Object Classification

Initial experiments on object recognition using local curve matching [3] gave promising results, and we plan to continue and develop more robust versions. Currently we have developed an algorithm which computes the distance between two contours of possibly rather different shapes. The method is local and fast, relying on both local similarities and global alignment. We describe extensive experiments with real images of $3D$ objects obtained from different viewing positions, as well as curves under partial occlusion, and curves describing different objects. In addition, we used the

method to compare a range of curves taken from a large database of real images of various toy models, some of them very different, others rather similar. We then used the results to classify the data with an automatic hierarchical clustering algorithm, getting excellent results which faithfully captured the real structure in the data.

8 Demonstrations and Evaluation

The following are demonstrations that we expect to be able to perform in the early phase of the program:

- Demonstrate the creation of panoramic scene manifolds from a video taken by a camera undergoing general motion and in a wide variety of scenes. While initial implementation will be running off-line on recorded video, it is expected to be running real-time, without hardware acceleration, towards the end of the project. An example of such a panoramic manifold is displayed in Fig. 1. While this initial display is done indoors, the demonstration is expected to be indoors as well as outdoors.

  The manifolds will be evaluated regarding their quality (sharpness and distortion), and by the accuracy they can add to alignment of objects.

- Demonstrate the creation of synthetic views of a scene from a small number of images taken from few views. While initial experiments are demonstrated on small objects in Fig. 2, it is expected that this approach will be applicable to indoor and outdoor scenes, and maybe even to scene manifolds.

  New views will be evaluated by comparing a simulated new view to a real image taken at the appropriate imaging conditions.

- Detecting and tracking multiple object will be demonstrated from a video recorded by aerial and ground moving sensors. Evaluation will be done by counting the mistakes in the detection and in the tracking. It is expected that we could display the tracking on the image manifolds to enable the operator to view the context as well as the tracked objects.

9 Bibliography

[1] S. Avidan and A. Shashua. Novel view synthesis in tensor space. In *CVPR-97*, San Juan, Puerto Rico, June 1997. Also http://www.cs.huji.ac.il/~shashua.

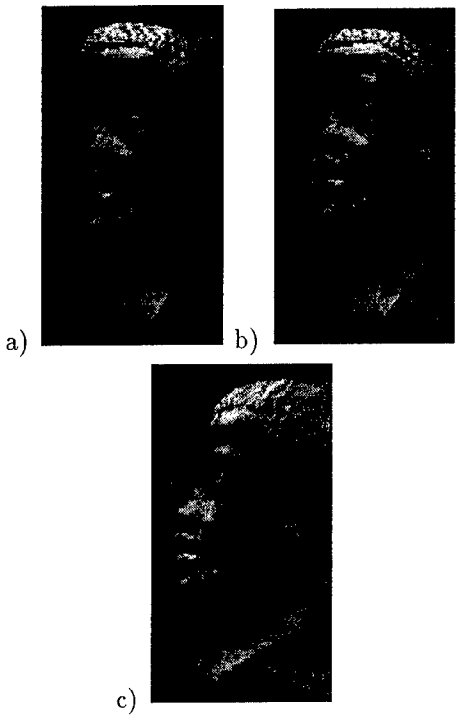


Figure 2: Generation of new views. (a)-(b) are two original views. (c) is a synthesized new view.

[2] S. Avidan and A. Shashua. Unifying two-view and three-view geometry. In *IUW-97*, New Orleans, Louisiana, May 1997. Morgan Kaufmann.

[3] Y. Gdalyahu and D. Weinshall. Local curve matching for object recognition without prior knowledge. In *IUW-97*, New Orleans, Louisiana, May 1997. Morgan Kaufmann.

[4] G. Halevi and D. Weinshall. Motion of disturbances: Detection and tracking of multi-body non rigid motion. In *CVPR-97*, San Juan, Puerto Rico, June 1997.

[5] M. Irani, P. Anandan, , and S. Hsu. Mosaic based representations of video sequences and their applications. In *ICCV*, pages 605–611, Cambridge, MA, June 1995. IEEE-CS.

[6] M. Irani and P. Anandan. Parallax geometry of pairs of points for 3D scene analysis. In *ECCV-96*, pages I:17–30, Cambridge, UK, April 1996. Springer.

[7] M. Irani and P. Anandan. A unified approach to moving object detection in 2D and 3D scenes. In *ARPA Image Understanding Workshop*, pages

Figure 1: A panoramic images using manifold projection. The curved boundary is created by the unstabilized motion of the hand-held camera.

707–718, Palm Springs, California, February 1996. Morgan Kaufmann.

[8] M. Irani and P. Anandan. Robust multi-sensor alignment. In *IUW-97*, New Orleans, Louisiana, May 1997. Morgan Kaufmann.

[9] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, 1994.

[10] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. In *CVPR-94*, pages 454–460, Seattle, WA, June 1994.

[11] S. Peleg and J. Herman. Panoramic mosaics with videobrush. In *IUW-97*, New Orleans, Louisiana, May 1997. Morgan Kaufmann.

[12] Y. Rosenberg and M. Werman. Representing local motion as a probability distribution matrix and object tracking. In *IUW-97*, New Orleans, Louisiana, May 1997. Morgan Kaufmann.

[13] B. Rousso, S. Avidan, A. Shashua, and S. Peleg. Robust recovery of camera rotation from three frames. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, California, June 1996.

[14] B. Rousso, S. Peleg, and I. Finci. Mosaicing with generalized strips. In *IUW-97*, New Orleans, Louisiana, May 1997. Morgan Kaufmann.

[15] A. Shashua. Algebraic functions for recognition. *IEEE-PAMI*, 17:779–789, 1995.

[16] A. Shashua and S. Avidan. The rank4 constraint in multiple view geometry. In *ECCV-96*, pages II:196–206, Cambridge, UK, April 1996. Springer.

[17] A. Shashua and S. Maybank. Degenerate n points configurations of three views: do critical surfaces exist? Nov. 1996.

[18] A. Shashua and N. Navab. Relative affine structure: Theory and application to 3D reconstruction from perspective views. In *CVPR-94*, pages 483–489, Seattle, WA, June 1994.

[19] A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In *ICCV*, pages 920–925, Cambridge, MA, June 1995. IEEE-CS.

[20] G. Stein and A. Shashua. Direct methods for estimation of structure and motion from three views. In *IUW-97*, New Orleans, Louisiana, May 1997. Morgan Kaufmann. To Appear in CVPR'97.

[21] D. Weinshall, M. Werman, and A. Shashua. Duality of multi-point and multi-frame geometry: Fundamental shape matrices and tensors. In *ECCV-96*, pages II:217–227, Cambridge, UK, April 1996. Springer.

Project Plan for Multiple Perspective Interactive Video Surveillance and Monitoring

Ramesh Jain
Visual Computing Laboratory, Electrical and Computer Engineering
University of California at San Diego, La Jolla, CA 92093-0407
E-MAIL: jain@ece.ucsd.edu
HOMEPAGE: http://vision.ucsd.edu/

Abstract

The Multiple Perspective Interactive Video (*MPI-Video*) project has been active for more than two years and has already demonstrated its applicability in areas including video surveillance and monitoring (VSAM). *MPI-Video* is an infrastructure for the analysis, management and interactive access to multiple streams of video cameras monitoring a dynamically evolving scene. Our *MPI-Video* VSAM (MPI-VSAM) project will develop technology to support crucial data processing activities. Specifically, we will design and implement robust and comprehensive methods for assimilating information coming from a multitude of distributed real-time sensors. Our information assimilation activities are formalized in terms of a Battlefield Environment Model (BEM), an extension of information processing research currently being done at our lab. The BEM represents a *Gestalt* of all the information obtained from individual sensors, and thus reflects a synergistic integration of a variety of information sources to construct a comprehensive, dynamically evolving model of the world. This model is used to provide strong contextual guidance for image understanding algorithms, such as motion detection, tracking, and activity understanding.

1 Introduction

The Multiple Perspective Interactive Video (*MPI-Video*) project has been active for more than two years and has already demonstrated its applicability in areas including video surveillance and monitoring (VSAM). *MPI-Video* is an infrastructure for the analysis, management and interactive access to multiple streams of video cameras monitoring a dynamically evolving scene [Jain and Wakimoto, 1995, Kelly *et al.*, 1995]. It has dominant database and hypermedia components which allow a user not only to interact with live events, but to browse the underlying database for similar or related events. The interactive construction of queries is also supported.

For video surveillance and monitoring large areas, sensor data from many platforms must be analyzed in a unified manner. Since a battlefield or any important urban site is too large to be covered just by one camera, it is essential that multiple platforms be used to acquire data from multiple perspectives. This system should be operational, independent of the time of the day and the season, requiring different types of sensors. The system composed of all these sensors mounted on multiple platforms should function in unison and present a *Gestalt view* to a user. Important research issues that must be addressed in this area include, assimilation of information from multiple sensors, determination of camera placement, dynamic scene segmentation, event understanding, camera handoff, and representation of individual sensor and global information. The details of all these issues are presented in [Boyd *et al.*, 1997].

Our *MPI-Video* surveillance and monitoring (MPI-VSAM) project will develop technology

to support these crucial data processing activities, so essential for successful VSAM systems. Specifically, we will design and implement robust and comprehensive methods for assimilating information coming from a multitude of distributed real-time sensors. Our information assimilation activities are formalized in terms of a Battlefield Environment Model (BEM), an extension of information processing research currently being done at our lab. The BEM is a time-varying model that provides its users up to date information about an extended and dynamically evolving environment covered by stationary and moving video and infrared cameras. The BEM represents a *Gestalt* of all the information obtained from individual sensors, and thus reflects a synergistic integration of a variety of information sources to construct a comprehensive, dynamically evolving model of the world. This model is used to provide strong contextual guidance for image understanding algorithms, such as motion detection, tracking, and activity understanding, issues of primary concern in our current research efforts.

2 Research Issues

There are several research issues related to image understanding, databases and informations systems, and network and storage architectures in the *MPI-Video* project. The MPI-VSAM project will focus on issues related to image understanding only. These issues are briefly discussed here. For details see [Boyd *et al.*, 1997].

2.1 Assimilation of Information

Information from multiple platforms must be assimilated into the central environment model. This model will have the *Gestalt* view of the environment. Also, information in this model will be represented at multiple levels of resolution. In addition to transformation of information from one abstraction to the other in the environment model, techniques must be developed to combine imprecise and unreliable information coming from these sensors. We will adopt our current EM to represent this for the VSAM scenario. We have started formulating information assimilation problem and soon we intend to start experiments in this area.

2.2 Motion Detection and Tracking

The environment model contains up to date information about objects in the environment. This model can provide a strong context for motion detection and tracking to individual sensors. We will adopt existing techniques to utilize the context provided by the EM to improve the motion detection and tracking algorithms. In particular, we are developing a mixture density based approach for segmentation of dynamic scenes and for tracking objects to identify motion parameters that will help determine events.

2.3 Tracking Across Multiple Platforms

Techniques will be developed to track objects as they move in the environment. Objects may go from one platform's scope to another and must be tracked smoothly and reliably as they cross platform boundaries. The user should be shown views from the environment model, independent of the camera view. Also, this form of tracking may require use of different sensors and steerable cameras to follow an object efficiently.

2.4 Activity Understanding and Discrimination

Techniques for defining complex activities, such as the discrimination of military incursions from common civilian activity or increased activity at an airport, in terms of simpler activities in the environment will be developed. The simpler spatio-temporal activities should be detectable in the environment model based on data acquired from individual platforms. We will develop algorithms to detect these activities from image data using activity recognition algorithms which employ Hidden Markov Model (HMM) techniques developed in our group.

2.5 Best and Virtual Views

Often an operator is interested in looking at the best view of an object or event. In this case, algorithms should be developed to provide the operator with the best view by switching to the appropriate camera. The best view determination will require definition of the best view, and optimization and switching criteria for selecting the best camera at each moment in time.

It is possible to synthesize virtual views of the environment using the environment model and the data available from sensors. We will develop and provide algorithms to generate these virtual views so that an operator may see an object of interest from an arbitrary view.

3 Evaluation

In the proposed effort, we will employ ten to twenty video and infrared sensors to allow day and night operations. These sensors will be mounted in and around our building to construct a realistic urban surveillance laboratory. This is a straightforward extension of our existing testbed that uses six video cameras. The set-up will be used to stage and record different events of interest. The effectiveness of our assimilation system will be evaluated for various conditions such as different environmental conditions, different population density (a few people, large crowds) and different activities or behavior of dynamic scene objects. Our evaluation will assess how well our system detects, locates and tracks moving people and vehicles in this area under the conditions mentioned above. Furthermore, we will evaluate the event detection and activity understanding capability of the system. This assessment will be in terms of how well the system can discriminate between different group or individual behaviors, as exhibited by the volunteers participating in these live tests.

Many different types of objects appear in the area where we will mount cameras. These objects are usually people, bicycles, golf carts, and some small trolleys and carts. The number of objects and activity depends on the time of the day and varies significantly. For example, when classes change on our campus, hundreds of students are seen walking, while in between classes, there is little activity. These variations in activity will provide us a natural setting to test activities of different kinds in a natural setting. We will also stage activities at specific locations in the field to compare them with the quantitative results obtained by the system.

A very important fact about our set up is that these cameras will be mounted in an actual area where all the activity takes place. This area will be our laboratory. We will record different scenes and then run experiments in our lab. For each subtask, we will define several quantitative experiments to evaluate our algorithms.

References

[Boyd *et al.*, 1997] J. E. Boyd, E. Hunter, R. Jain, P. H. Kelly, J. Schlenzig, and A. Tai. Multiple perspective interactive video surveillance and monitoring. In *1997 DARPA Image Understanding Workshop*, New Orleans, May 1997.

[Jain and Wakimoto, 1995] Ramesh Jain and Koji Wakimoto. Multiple Perspective Interactive Video. In *Proceedings of International Conference on Multimedia Computing and System*, Washington, D. C., USA, May 1995.

[Kelly *et al.*, 1995] Patrick Kelly, Arun Katkere, Don Kuramura, Saied Moezzi, and Shankar Chatterjee. An architecture for multiple perspective interactive video. In *Proceedings of ACM Multimedia 95*, 1995.

Image Understanding at Cornell University*

Daniel P. Huttenlocher Ramin Zabih

Computer Science Department
Cornell University
Ithaca, NY 14853
dph@cs.cornell.edu rdz@cs.cornell.edu

Abstract

This project focuses on tracking and recognition of vehicles and vehicular activity. We have developed a variety of methods that will be applied to this task, based primarily on non-parametric statistical methods. A central goal is to combine motion, color and 2D shape information in tracking, classification and recognition. Contextual constraints will also play a role, particularly in tracking. These systems will be demonstrated and evaluated by deploying them on the Cornell campus, for tasks where the performance can be independently verified.

1 Introduction

Our research addresses the task of tracking and recognizing vehicles and vehicular activity, both from ground-based and aerial platforms. A central focus is the use of non-parametric statistical methods, based on fundamental properties such as rank ordering. We also plan to develop methods that exploit contextual constraints in order to obtain more reliable and faster performance. For example, the motion of vehicles along roadways is highly constrained. This can be used to predict subsequent views of an object, to assist in tracking and to increase the speed and reliability of classification and recognition. The planned systems will build on methods that we have previously developed for shape- and color-based matching and tracking [Huttenlocher *et al.*, 1993b, Pass and Zabih, 1996]. We will also develop new methods that use motion in combination with spatial and color cues. Our goal is to develop cost-effective systems that run quickly on standard computing platforms with a few processors.

Techniques from robust statistics have been of considerable utility in developing computer vision algorithms that are reliable over a wide range of image conditions (e.g., [Besl *et al.*, 1988]). We have found non-parametric statistical measures to be particularly valuable, for problems including shape matching (using the Hausdorff distance [Huttenlocher *et al.*, 1993a]) and determining visual correspondence (using census transform correlation [Zabih and Woodfill, 1994]). A key property of non-parametric measures is that the outliers, or bad data, do not need to be explicitly modeled. There is only a limit on the number of outliers that can be tolerated, not on the form that they take. Such tolerance of unknown and variable data will enable the planned systems to operate at any time of day, in different weather conditions, and with unknown objects in the field of view (perhaps partly occluding objects of interest).

2 Context-based vehicle tracking

Our approach to tracking is based on matching two-dimensional image views in order to determine where an object at one time frame

*Supported by DARPA under Airforce contract 95040-6386 to Alphatech, and contract DAAH04-93-C-0052 to Hughes Aircraft.

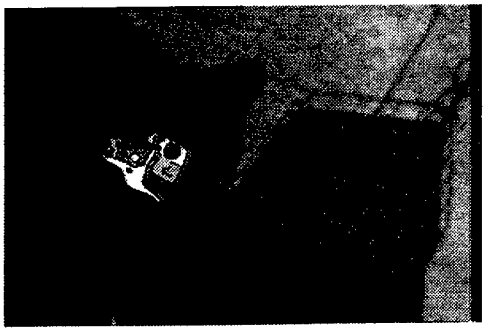


Figure 1: An example of vehicle tracking through heavy shadow regions, which change as the vehicle moves.

moved to in a subsequent frame [Huttenlocher *et al.*, 1993b]. Unlike other work on model-based tracking (e.g., [Lowe, 1992]), our method is not restricted to tracking rigid or articulated objects. Moreover, our approach does not require a geometric model of an object before it can be tracked. This is particularly important for video monitoring, where the range of possible objects is large and where there are many possible distractors that may also need to be tracked. We have demonstrated matching-based tracking mechanisms which work well even in situations where objects becoming temporarily hidden from view or pass through strong shadow boundaries [Huttenlocher *et al.*, 1993b], as illustrated in Figure 1. One of the first challenges that our new research is addressing is maintaining high reliability tracking for considerably lower-resultion objects than that pictured here.

We plan to use a three-stage process for tracking vehicles: *segmentation*, in which we identify regions of the image that contain moving objects; *discrimination*, in which we ascertain whether each moving object is a vehicle, and not, e.g., a pedestrian or animal; and *following*, in which we track an object as it follows a path (generally along a roadway).

Segmentation. Segmentation will be done by distinguishing between background motion and object motion. For non-rigid objects, simple techniques may result in one object being split into several moving pieces. In tasks that involve tracking pedestrians, we will develop techniques that use local time intervals to obtain more reliable estimates of what is moving together. For vehicles, simple techniques based on differential motion should suffice. Because the objects of interest move at a wide variety of speeds, this processing will need to be done at several different time scales. For vehicle tracking it is particularly important to find the bottom of the object, as this indicates its position on the ground plane.

Discrimination. It is necessary to determine if a moving object is a vehicle or something else (pedestrian, animal, etc.). We will exploit several contextual constraints that are specific to the task of vehicle identification on roadways, as follows. *Position and motion direction:* Vehicles will not in general drive on the sidewalk. Pedestrians can be in the street temporarily, but when there are typically moving perpendicularly to the flow of traffic. *Size:* Vehicles are substantially larger than pedestrians, especially in width and length. *Rigidity:* Vehicles undergo rigid motion, while living creatures typically are highly non-rigid. We will construct an algorithm to detect whether or not an object is rigid. *Speed and motion smoothness:* Vehicles generally move faster than pedestrians, and their motion forms smooth trajectories, with substantial smooth accelerations and decelerations.

Following. We will use a model-based tracker, based on the work described in [Huttenlocher *et al.*, 1993b]. This tracker matches a two-dimensional model from the previous time frame to a region of the current image frame, in order to determine where the object being tracked is located at the current time. We will use contextual constraints in order to speed up the operation of the tracker, and potentially to also make it more reliable. One powerful constraint is the fact that a vehicle should move along a roadway, so the image search is really just one-dimensional. This greatly restricts the search region for matching. Moreover, the vehicle orientation with respect to the camera is given by the position along the roadway (assuming the roadway location and orientation are known in the camera coordinate frame). This can be used to transform the view from the previous time frame before performing a match. We will evaluate the speed and accuracy of the

tracker with and without such constraints.

We will also investigate techniques for determining when the contextual constraints used by the tracker have been violated, and therefore should be relaxed to enable tracking to continue. One approach is to expand the search window when the tracker fails to find a match in the window that was specified using the contextual constraints. If a match is then found outside the specified search window, this is evidence that the contextual constraints are being violated. If this continues for several frames, the tracker can be adjusted to use the new more relaxed constraints.

2.1 Classification and recognition

In the planned systems, there are both *classification* and *recognition* problems. By classification, we mean the problem of determining which objects are instances of the same class. By recognition (or identification) we mean finding the best match between an unknown object and a set of object models, in effect assigning a name to the unknown object. Classification can be performed without the need for prior models, while recognition problems involve prior models. In convoy detection, classification is sufficient for the problem of determining which vehicles are instances of the same vehicle type. On the other hand, counting the number of vehicles of a given type that pass through an intersection may require recognition (the ability to search for a given type of vehicle in the image).

We plan to use a view-based approach to classification and recognition, where an object is represented by a sequence of two-dimensional views. These sequences will be compared using view matching techniques based on the Hausdorff measure [Huttenlocher *et al.*, 1993a]. While three-dimensional information (e.g., from stereo) could be useful for estimating the size of objects, we will instead use contextual constraints (e.g., by exploiting information about the scene geometry of roadways). For both classification and recognition, we plan to use multiple views of a vehicle as it passes by an observation point (for some camera positions this can include front, side and rear views). The use of multiple views makes it possible to classify and recognize objects despite transient problems such as specular reflections (which generally occur over a small range of views), glitches in the video signal, or when a vehicle is occluded from view by a passing vehicle. Multiple views also enable the use of those views that best distinguish a given object from other objects (e.g., an APC and a tank are similar except from frontal and side views where the turret and barrel are visible).

Both classification and recognition require similarity measures for comparing views. For reliability, we will use quite different measures, based on shape and on color. We will use the Hausdorff fraction [Huttenlocher *et al.*, 1993a] for shape, which measures the portion of one binary image that approximately overlaps with another. This measure is robust to both partial occlusion and to positional uncertainties in feature locations. Our approach to color is based on computing joint histograms of local image properties (see section 4).

The classification problem involves partitioning a set of objects into subsets of equivalent objects (classes). We will use standard clustering techniques to form such equivalence classes. Since there are multiple views of each object, and the correspondence of views is known, the clustering process should make use of the information from multiple views in an intelligent manner. Suppose two objects match well in all views except one. This could be due to a transitory event such as a passing vehicle causing a false mismatch in one view, or it could be due to the fact that the two objects actually only differ in one view. While for a single pair of objects these two cases cannot be distinguished, for a set of objects there can be good heuristics for doing so. For instance, in convoy detection it is reasonable to assume that "singleton" objects (those which don't match any other object) are unlikely, as a convoy consists of repeated vehicles.

One approach to recognition that we are pursuing is based on extending the eigenspace approximation to the Hausdorff fraction [Huttenlocher

et al., 1996]. This technique uses a subspace approach to enable a large number of stored model views to be efficiently matched against an unknown view. Unlike other subspace methods which are based on the SSD and thus sensitive to outliers, this method uses the robust Hausdorff fraction for matching. We will develop a system for recognizing view sequences, which makes use of known viewpoints for some or all of the views (as discussed above) to speed the image matching process. We will also investigate new matching measures for robustly combining information from multiple views of an unknown object. One approach is to discard outlying views of an unknown object that don't match any stored model views well, and then to vote with the remaining views. This prevents bad views (due to occlusion, specularities, etc.) from throwing off the matching process.

2.2 Evaluation plan

The planned systems were chosen in part because they have independently verifiable outputs, such as the number of vehicles appearing in the field of view during a given time interval. This enables us to evaluate system accuracy by comparing it with human performance on the same task. Our goal is to achieve accuracy comparable to human observers (of course human accuracy will vary, depending on the observer and the conditions). System speed, and the tradeoff of speed versus accuracy will also be evaluated. Some of the modules that we will construct will also have verifiable outputs, and these modules will be similarly evaluated against observer-supplied ground truth. On the other hand, several of the modules do not have verifiable outputs. For instance, there is no clear definition of the "correct" motion segmentation for a frame. For such modules we will do informal tuning and evaluation. We will do formal evaluations, using ground truth, both for the complete systems and for the modules with verifiable outputs. The modules which we evaluate informally will play a vital role in the complete systems. They will therefore be subject to an indirect evaluation against ground truth as part of an overall system.

3 Comparing matching measures

We are interested in characterizing how various matching measures differ from one another in terms of their ability to correctly detect a distorted instance of a target in clutter. In order to determine the power of different measures, we use Monte Carlo techniques to estimate Receiver Operating Characteristic (ROC) curves for each measure. These curves give the tradeoff between probability of detection and probability of a false alarm for the different measures, thus enabling a determinination of which measures perform better under which operating conditions. We consider variations in the amount of occlusion of the target, the amount of background clutter, the type of background clutter (correlated noise such as "edge chains" versus uncorrelated noise such as points), and the spatial perturbation of the target feature points.

Thus far we have compared several measures for matching binary images that are based on distance transforms. A distance transform of a binary image defines for each image pixel the distance to the nearest "on" pixel of that image, using a given distance function such as Euclidean distance (the L_2 norm). In order to match two images, the "on" pixels of one image are used as *probes* to select distance transform values of the other image. Such measures have formed the basis of a number of model-based recognition techiques (e.g., [Huttenlocher *et al.*, 1993a, Paglieroni, 1992]), where they are used to compare binary attributes extracted from image data. The evaluation results are described in more detail elsewhere in these proceedings (see the paper by Huttenlocher).

4 Color-based image comparison

Color-based comparison of unmodeled objects is typically done via color histograms. Histograms are robust to large changes and viewpoint, and are trivial to compute; however, they fail to incorporate spatial information. We have developed a class of methods for combining color information with spatial layout, while retaining the advantages of histograms. Our approach is based on computing joint histograms of several

local properties. Joint histograms can be compared as vectors, just as color histograms can. However, in a color histogram any two pixels of the same color are effectively identical. With joint histograms, pixels must share several properties beyond color. We have explored a number of different local spatial properties. For example, we can divide pixels into classes based on their spatial coherence and (coarse) position in the image. We demonstrate in [Pass and Zabih, 1996] that these simple, efficient measures perform significantly better than color histograms, especially when the number of images is large. We have also investigated the distribution of a given color as a function of the distance between two pixels. The resulting method, which we call a *color correlogram*, has also proven to be quite effective. See the paper by Huang and Zabih in these proceedings for more details.

5 Adaptive-window methods

Early vision relies heavily on rectangular operators for tasks such as smoothing, segmentation or computing correspondence. While rectangular windows have obvious efficiencies, they poorly model the boundaries of real-world objects. We have developed an efficient method for adaptively choosing a window without a rectangular bias, in a manner which varies at each pixel. We model an image as a piecewise constant function corrupted by noise, and explicitly consider all possible connected components. Almost all components can be pruned, however, by a simple maximum likelihood argument. The remaining components can be compared by a variety of methods, including (for example) global contextual constraints. Our approach can be applied to many problems, including image restoration, motion and stereo. It can help solve a number of well-known problems, including a version of the aperture problem. Our methods run in a few seconds on traditional benchmark images with standard parameter settings, and give quite promising results. Details are given in the paper by Boykov, Veksler and Zabih in these proceedings.

References

[Besl *et al.*, 1988] Paul Besl, Jeffrey Birch, and Layne Watson. Robust window operators. In *2nd International Conference on Computer Vision*, pages 591–600, 1988.

[Huttenlocher *et al.*, 1993a] Daniel Huttenlocher, Greg Klanderman, and William Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[Huttenlocher *et al.*, 1993b] Daniel Huttenlocher, Jae Noh, and William Rucklidge. Tracking nonrigid objects in complex scenes. In *4th International Conference on Computer Vision*, pages 93–101, 1993.

[Huttenlocher *et al.*, 1996] Daniel Huttenlocher, Ryan Lilien, and Clark Olson. Object recognition using subspace methods. In *4th European Conference on Computer Vision*, volume 1, pages 536–545, 1996.

[Lowe, 1992] David Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, 1992.

[Paglieroni, 1992] D.W. Paglieroni. Distance transforms: Properties and machine vision applications. *Computer Vision, Graphics and Image Proc.: Graphical Models and Image Processing*, 54(1):56–74, 1992.

[Pass and Zabih, 1996] Greg Pass and Ramin Zabih. Histogram refinement for content-based image retrieval. In *IEEE Workshop on Applications of Computer Vision*, December 1996.

[Zabih and Woodfill, 1994] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *3rd European Conference on Computer Vision*, pages 151–158, 1994.

Image Understanding Research at CMU

T. Kanade and K. Ikeuchi

School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213
tk@cs.cmu.edu
http://www.cs.cmu.edu:8001/afs/cs/project/vision/www/vasc.html

Abstract

The CMU Image Understanding program continues to comprise a panorama of activities ranging from basic vision research to the development of new application systems. This PI report summarizes our recent activities in the following focal areas:

- Basic vision research
- Real-time vision
- Video surveillance and monitoring
- Mobile robot navigation
- Image and video database
- Image-Guided Surgery

Taken together, these represent progress on a broad front towards deployable, dependable and taskable machine vision systems.

1. Basic Vision Research

While many vision systems have been demonstrated in principle, few have been highly reliable when deployed. This failure is largely because of the reliance on vision modules which are based on oversimplified assumption of the principles of imaging process. To remedy this deficiency, we have pioneered at CMU the exploration of vision science, the careful analysis of each vision process based on law of physics and mathematics. Our work in this area has already demonstrated major advantages in deploying vision systems. We continue this effort to study the fundamental issues. In particular, this PI report highlights two fundamental vision issues: object representation and recognition in vision science.

1.1. Object Representations

One of the fundamental requirements in computer vision is the construction of object representations. These object representations are later utilized in various vision tasks such as recognition, tracking, visualization, and navigation. In the past, representations have typically been manually created using computer-aided design tools. Manual modeling, however, suffers from expense and accuracy limitations. Our work has been directed towards overcoming these limitations by automatically constructing object representations from real images of the object.

1.1.1. Geometric Representation

Object representations have two aspects: geometric and photometric. Geometric representations convey three dimensional shapes of objects, while photometric representations do appearances of objects, including surface textures, real color and reflective properties. We examine the methods to automatically acquire both of these representations from real images of objects. In this section, we

This research has been supported in part by the Defense Advanced Research Projects Agency under the Department of the Army, Army Research Office grant number DAAH04-94-G-0006. Views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the Department of the Army or the United States Government.

focus our recent achievement to acquire geometric representations.

1.1.1.1. Acquiring Surface representation

We (Wheeler, Sato and Ikeuchi) developed a system that creates 3D surface representations from range images of the object. Our basic approach is to acquire several range-image views of the object, align the image data, merge the image data using the aid of a volumetric representation, and then extract a triangle mesh from the volumetric representation of the merged data. Our main contribution is a new algorithm for computing the volumetric representation from the sets of image data [Wheeler, 1996]. Our algorithm, the consensus-surface algorithm, eliminates many of the troublesome effects of noise and extraneous surface observations in the data. It does so by searching for a consensus of surface observations in order to estimate the implicit distance from each point in the volume to the closest point on the surface. From the discrete implicit surface representation in the volumetric grid, it is straightforward to generate a smooth triangulation of the surface.

This algorithm can produce accurate object models despite the poor quality of data available from real imagery (for both range and intensity images). Our algorithms achieve robustness by searching for consensus information among several views to determine which image features justifiably constitute an element in the model. As an example, Figure 1 (a) shows a surface generated by a naive algorithm, while Figure 1 (b) shows those obtained by our algorithm.

1.1.1.2. Multi-Scale Representations

We (Zhang and Hebert) have developed an approach to representing shapes at different levels of resolution [Zhang and Hebert, 1996], to control appropriate details in terms of memory efficiency and efficient matching.

This approach starts with a smoothing algorithm for representing objects at different scales. In a way similar to the classical scale space representations, larger amount of smoothing removes more details from the surfaces. Smoothing is applied in curvature space directly, thus avoiding the usual shrinkage problems and allowing for efficient implementations. See Figure 2.

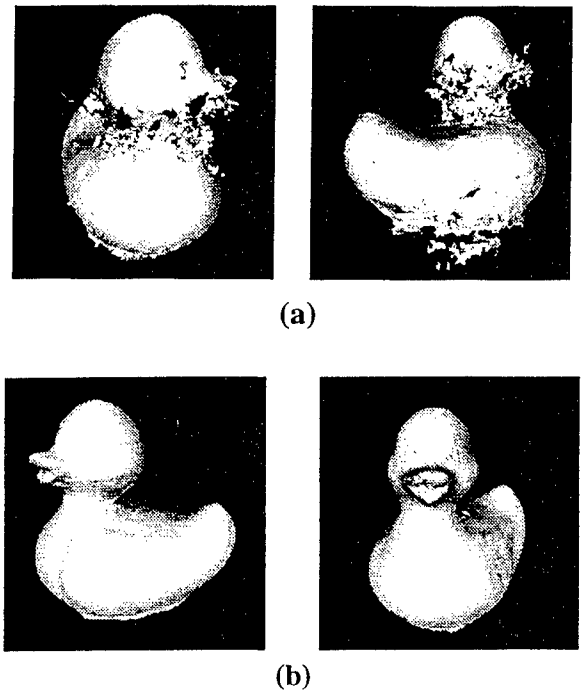


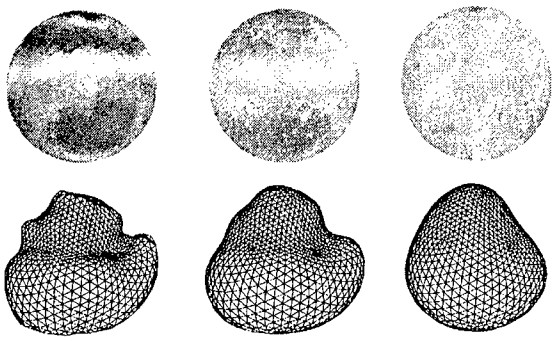
Figure 1: Models generated (a) a naive algorithm and (b) our algorithm

We introduced a 3D similarity measure that integrates the representations of the objects at multiple scales. Given a library of models, objects that are similar based on this multi-scale measure are grouped together into classes. We showed how shapes in a given class can be combined into a single prototype object by using the technique of inverse mapping from representation to shape, introduced in our earlier work.

Finally, the derived prototypes are used for hierarchical recognition. The input scene representation is first compared to the prototypes and then matched only to the objects in the most likely prototype class rather than to the entire library of models. Beyond its application to object recognition, this approach provides an attractive implementation of the intuitive notions of scale and approximate similarity for 3D shapes.

1.1.2. Photometric Representation

Generation of object models requires two pieces of information: the object's shape (geometric information) and reflectance properties (photometric



(a) (b) (c)

Figure 2: Smoothed curvature distribution (top) reconstructed mesh by inverse mapping (bottom) for different degree of smoothing, σ. (a) σ=0.5; (b) σ=2.0; (c) σ=5.0.

information) such as color and texture. The previous section describes one half of this effort. This section describes our effort on the second half.

1.1.3. Acquiring Photometric Representations

We (Sato and Ikeuchi) have been developing a method to acquire photometric representations - the color, texture, and reflectance parameters - of an object by observing a real object. Previously, we developed a method to analyze a sequence of color images taken under a moving light source [Sato and Ikeuchi, 1994]. The method, goniochromatic space analysis (GSA), allows us to estimate the photometric information of the object model - the diffuse and specular reflection components - from a series of color images of an object.

In our recent work [Sato and Ikeuchi, 1996, Sato, 1997], GSA has been extended to handle moving object.s. First, a sequence of range and color images of a real object is measured by rotating the object on a rotary table. Then, the object shape is obtained as a collection of triangular patches by merging multiple range images. Second, a sequence of color images is mapped onto the recovered shape. As a result, we can observe color changes for each triangular patch of the object surface throughout the image sequence. The observed color sequence is separated into the diffuse reflection component and the specular reflection component by the modified GSA. A linear combination of the Lambertian model and the Torrance-Sparrow model is used as the reflection model.

For accurately recording texture distribution of an object, we divide each triangular patch into finer grids. For each grid, a set of pieces of information consisting of body color, specular parameters, and the surface orientation are stored. Once we have this representation of an object, we can synthesize very realistic images using the stored parameter-sunder various illumination condition.

1.1.4. Acquiring 3D Edgel Representations

Three-dimensional (3D) edgel representations are useful for object recognition. These 3D edges are due to various visual effects: orientation discontinuities, occluding boundaries, and surface texture. Thus, it is quite difficult to predict edgel appearances.

We (Wheeler and Ikeuchi) have developed a system that automatically acquires a 3D edgel model of an object from images. First, a 3D model of the surface is built using the previously described 3D modeling system. A set of intensity-image views of the object is collected, and the edgels from them are extracted using a standard edge operator. The edgels are then projected and aligned in the object's 3D coordinate system using a 3D surface model of the object (built using the 3D surface modeling approach alluded to previously). The aligned 3D edgel data is then merged to produce a set of rigid edgels belonging to the object. We again use the concept of consensus to perform the merging --- extracting statistically significant/ salient 3D edgels from the collection of observed 3D edgels. To account for occluding-contour edgels, we use curvature analysis of the points on the 3D surface model to predict which surface points are contour edgel generators.

The experimental results demonstrate that our modeling algorithms can extract clean models from rather noisy data, and that these models can be efficiently and effectively used for localization tasks.

1.2. Object Recognition

We have devoted a substantial amount of effort toward robust and general object recognition. First, we have addressed two major problems in object recognition: object localization and object classification. We have built and successfully demon-

strated two object recognition systems. The first uses a novel approach to 3D surface representation to recognize objects in complex scenes using range data; the second one recognizes objects in SAR images.

1.2.1. Robust Algorithms for Object Localization

Being able to accurately estimate an object's pose (location) in an image is an important and practical problem. Recognition algorithms often trade off accuracy of the pose estimate for efficiency -- usually resulting in brittle and inaccurate recognition. One solution is object localization -- a local search for the object's true pose given a rough initial estimate of the pose.

We (Wheeler and Ikeuchi) have been developing algorithms for object localization [Wheeler and Ikeuchi, 1993] [Wheeler, 1996]. In previous work, we developed and tested a robust algorithm for localizing 3D (arbitrary-shaped, rigid) objects in range images (3D-3D localization). Our localization algorithm iteratively refines the pose by optimizing an objective function defined over the image data, model data and the object's pose. The feature of this work was the use of a robust objective function to reduce the effect of noise and outliers which are prevalent in real image data. This method proved capable of efficiently and accurately localizing 3D objects in range images despite clutter and significant amounts of missing and occluded data.

In the past year, we have extended the technique to localizing 3D objects in 2D intensity images using edge matching [Wheeler, 1996]. In order to deal with intensity images, we developed a new representation of an object's appearance as a collection of 3D edgels (edge points or elements). By representing a 3D object as a dense collection of points on the surface which are capable of generating intensity edges in images, we have a way to connect the 3D object (and its pose) to the 2D image observations (edges). Edgels provide a very general representation since any shape can be approximated as a collection of points. Our representation also allows for 3D edgels which generate occluding contours. Hence, our representation can account for a wide variety of rigid object shapes (smooth and non-smooth).

The algorithm has been tested with a wide variety of objects. Despite noisy image data and large initial pose errors (10 mm and 15 degrees error for objects of scale 100 mm), it successfully localized objects correctly. In addition, we showed that multi-image localization (implicit model-based triangulation without computing depth) was shown to provide substantial improvements in the resultant pose estimation.

1.2.2. Recognition of 3D Objects for Remote Operation Applications

For recognition of complex 3-D objects in range images, we (Johnson and Hebert) have developed a representation that combines the descriptiveness of global object properties with the robustness to partial views and clutter of local shape descriptions [Johnson and Hebert, 1997]. A local basis is computed at an oriented point (3-D point with surface normal) on the surface of an object. All the positions on the object surface now can be described with respect to the basis of other points by two parameters. By accumulating these parameters in a 2-D array, a descriptive image (*spin-image*) associated with the point is created. Because spin-images describes the coordinates of points on the surface of an object with respect to the local basis, they are local encoding of the global shape of the object and are invariant to rigid transformations.

To prepare a model for recognition, a spin-image is generated for each vertex in the model mesh. The top images in Figure 3 show some representative spin-images for a model of a valve.

At recognition time, spin-images from points on the model are compared with spin-images from points in the scene; when two images are similar enough, a point correspondence between model and scene is established. After point matching, a model is localized in the scene by grouping correspondences to compute a transformation which is subsequently refined and verified using a modified iterative closest point registration algorithm.This recognition algorithm has been integrated into a semi-automatic world modeling system called Artisan [Johnson et. al., 1997b]. Artisan combines

3-D sensors, object modeling and analysis software, and an operator interface to create a 3-D model of a robot's work area. Through object recognition, Artisan assigns semantic meaning to objects in the scene which facilitates execution of robotic commands and drastically simplifies operator interaction. Figure 3 shows the recognition of a valve model in a complex scene typical of interior work environments. Artisan was demonstrated in several tasks at the Oakridge National Labs, using a remotely operated mobile platform.

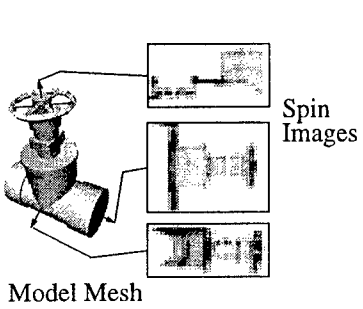


Figure 3: Recognizing objects in complex scenes. This technique is used in the Artisan system for remote maintenance.

1.2.3. Object Recognition in SAR Images

Automatic target recognition (ATR) using synthetic aperture radar (SAR) images is a promising military application area, because SAR sensors allow continuous day/night coverage under all weather conditions, and can achieve high spatial resolution even from orbital platforms. Despite these advantages, SAR-based ATR systems are difficult to develop [Ikeuchi et al., 1996]. For objects with complex surface geometry, such as a tank, slight differences in viewing angle cause features to suddenly appear, disappear, and abruptly change shape. Furthermore, signals that bounce off multiple surfaces before returning give rise to non-attached features that appear to be floating beyond the target surface. Some prominent image features are sometimes useless for recognition.

The eigenspace method is a promising classification technique that can take a training set of target images and automatically determine what features are most important for recognition, and it is therefore a potentially useful approach to SAR ATR. However, there are several known drawbacks to the standard eigenspace method, in particular, it is sensitive to the placement (translation) of the object in the image, and does not handle occlusions and articulated object well.

We overcome these difficulties in eigenspace classification techniques by building a SAR ATR system based on small "eigenwindows" [Ohba and Ikeuchi, 1997]. This system divides each training image into small subwindows, all of which are stored as points in the eigenspace. An unknown target image is also broken into subwindows and projected into eigenspace. Each pairing of a target eigenwindow point and a training point votes for a particular target and viewing angle, and the final classification is achieved as the consensus of all such votes. This eigenwindow approach has a number of benefits. First, when some parts of a target are occluded, remaining windows covering visible parts can identify the target. Second, to detect a target with articulated components, we can define separate windows for each, and recognition can proceed separately on the articulated parts and the body. Third, the method is by definition insensitive to image translation. Finally, using multiple small windows rather than a whole image greatly reduces the dimensionality of the eigenspaces that must be manipulated.

We (Collins, Wheeler, Ohba and Ikeuchi) have implemented an eigenwindow-based SAR ATR system, and evaluated it using seven targets types: BMP, BTR60, KTANK, M35, M113, M60 and SCUD. Training images for each target were generated via the XPATCH simulator by varying the azimuth angle from 0 to 359 degrees in 1 degree

increments, while maintaining a constant SAR depression angle of 22.5 degrees and resolution of 30 cm/pixel. Test images were also generated via XPATCH, at fractional azimuth values. A target classification produced by the system was considered to be correct if it was of the correct object type, and had an estimated azimuth angle within 5 degrees of the correct angle. Under this criteria, when the system was tasked to produce a single, best candidate hypothesis, the mean classification accuracy was 95% (std of 4%) for unoccluded targets, and 93% (std 5%) for targets occluded up to 50% in the worst case.

The system was also tested in an indexing mode, where the top 5 best candidate hypotheses were generated. Under this tasking mode, the correct classification was contained in the top 5 hypotheses 99% (std 1%) of the time for unoccluded targets, and 98% (std 2%) for targets occluded up to 50%. System performance is therefore excellent, particularly when used as an object indexing system, and performance degrades slowly with respect to target occlusion.

2. Low-Latency Real-Time Image Processing

Latency, or reaction time, is the time that a system takes to react to an event. The primary sources of latency in vision systems are: the data transfer bottleneck caused by the need to transfer an image from the camera to the processor, and the computational load bottleneck caused by the processor's inability to quickly handle the large amount of visual data. The detrimental effects of both bottlenecks scale up with the image size.

Another aspect presently missing in machine vision is top-down sensory adaptation. Complex adhoc algorithms that try to extract relevant information from inadequate sensor data are inevitably unreliable. In fact, time and time again it has been observed that using the most appropriate sensing modality or setup, allows recognition algorithms to be far simpler and more reliable. A system that can adjust its operation at all levels, even down to the point of sensing, would be far more adaptive than the one that tries to cope with the variations at the "algorithmic" or "motoric" level alone.

We (Brajovic, Amidi, Madison and Kanade) have been developing low latency and adaptive real-time image processors based on a computational sensor paradigm and a reconfigurable 2D vision machine architecture.

2.1. Computational Sensors

The computational sensor paradigm [Kanade and Bajcsy, 1993] has the potential to greatly reduce latency and provide top down sensory adaptation to the vision system. By integrating sensing and processing on a VLSI chip both transfer and computational bottlenecks can be alleviated: on-chip routing provides high through put transfer, while an on-chip processor could implement massively parallel fine grain computation providing high processing capacity which readily scales up with the image size. In addition the tight coupling between processor and sensor allows for efficient top down feedback that can control and adjust sensor for further acquisition based on the preliminary results of the processing.

Our recent work has been concerned with efficient implementation of global operations over a large groups of image data using computational sensor paradigm [Brajovic and Kanade, 1994]. The data are supplied optically by focusing an image (henceforth referred to as a retinal image) onto the array of photo detectors. A processor integrated within the chip, incrementally makes decisions based only on a few input data at a time. The problem is how to efficiently chose which few input data to route to the global processor at each given time. Recently we investigated two models: the sensory attention [Brajovic and Kanade, 1996], and the intensity–to–time processing paradigm [Brajovic and Kanade, 1997].

More details are presented in our MURI PI Reports in the same proceedings.

2.2. Reconfigurable Vision Machine Architecture

A practical real-time vision machine must performs low-latency, high-bandwidth, and versatile operations on an uninterruptable, overwhelming volume of image data. Image processing functions can be local or global, their operations can be uni-

form or non-uniform, and their control flow can be data-dependent or data-independent. A suitable architecture for vision varies depending on applications, We have to balance input bandwidth (i.e. data access requirements), processor bandwidth, and output bandwidth for different applications. Our strategy for developing a vision machine architecture stresses modularity, expandability, and simplicity in configuring a target machine, rather than blind "generality". It is not important how general a fixed machine is, but how quickly a specific economical machine can be configured for an application at hand.

We (Amidi, Kanade and Madison) have been developing a two-dimensional Reconfigurable Vision Machine (RVM) pipelined architecture. The RVM architecture integrates functional modules with a unified communication interface, each configured into an application-specific system to achieve the best performance with the lowest cost. In designing such a target system, processing and input/output bottlenecks at various pipe stages are overcome by vertically and horizontally extending the pipes with modular hardware. To support such a process of system configuration, we have developed software systems with which a designer can graphically interconnect appropriate modules, simulate the operation of the target system, identify the system bottlenecks for improvement before hardware realization of a target system.

The RVM architecture employs four types of functional modules: Function-specific Image Processing (FIP) Modules, Processor Modules, Junction Modules, and Bridge Modules. The FIP modules are hardware-oriented modules which perform certain image processing functions such as image capture, display, table lookup, and convolution. Processor modules are self-contained programmable processing units used for realizing capabilities that are best implemented by software for flexibility. Junction modules and Bridge modules provide unified high-speed communication links. Junction modules perform tasks such as data broadcasting, branching, and merging. Bridge modules link off-the-shelf computer systems, such as personal computers or DSP engines, to the configured system.

The reconfigurable hardware operates in conjunction with a suite of software libraries and tools. These include tools which run on the PC to create DSP processor code, to build loadable images of programs and data, and to communicate with the host-interface module. Software libraries have been developed for the host interface and C44 processors which allow fast and simple development of typical computer vision applications. An operating system on the host-interface provides facilities for debugging, for communication with other modules, and for communications with man-machine interfaces running on PCs.

The RVM architecture has been already applied to industrial inspection problem.

3. Video Surveillance and Monitoring

Carnegie Mellon University and David Sarnoff Research Center have recently started a joint, integrated feasibility demonstration (IFD) effort in the area of Video Surveillance and Monitoring (VSAM). The objective of the VSAM project is to develop automated video understanding technology for use in future urban and battlefield surveillance applications, where human visual monitoring is too costly, too dangerous, or otherwise impractical. Sample applications include building and parking lot security, monitoring restricted access areas in warehouses and airports, scanning urban battlezones for sniper activity, and performing reconnaissance on the battlefield. Technology advances developed under this project will enable one human operator at a remote host workstation to supervise a network of VSAM platforms (stationary, moving on the ground, or airborne), having multiple, steerable sensors operating in the visible and infrared bands for continuous day/night operations. The platforms will be mainly autonomous, notifying the operator only of salient information as it occurs, and engaging the operator minimally to alter platform operations.

The technical objectives to be achieved by the CMU/Sarnoff effort are: 1) cooperative surveillance by multiple ground and airborne sensors to seamlessly track moving targets as they enter and leave the field of views of individual sensors, or become temporarily occluded from one or more sensor viewpoints. 2) Scene-level representation of targets and their environment by integrating evolving visual, geometric, and symbolic sensor obser-

vations together with collateral scene data. 3) Active control of sensor parameters, sensor processing, and platform deployment in response to mission and task needs based on the evolving wide area representation. 4) Development of an experimental testbed that includes the sensors, hardware platforms, and software architecture needed to support data collection and experimental evaluation of VSAM technologies developed by the DARPA IU community.

More details are presented in the VSAM-IFD PI Report in the same proceedings.

4. Mobile Robot Navigation

There are three main thrusts to CMU's mobile robot activities: the DARPA-funded Unmanned Ground Vehicle program, NASA's Lunar Rover initiative, and DOT on-road research. All three share common elements of sensing and sensor interpretation, and all draw heavily on IU work and results.

4.1. Unmanned Ground Vehicles (UGV)

We (Hebert, Stentz and Thorpe) have extended the capabilities of the UGV systems demonstrated as part of the DARPA/OSD Demo II program [Hebert, 1996; Hebert et al., 1996] in four areas: advanced sensors, perception, planning, and teleoperation.

In the area of advanced sensors, we have investigated the use of three perception sensors for obstacle avoidance and navigation. First, we have modified the frame-rate stereo machine developed by Kanade for use in UGV applications. The stereo machine was integrated in the navigation system and demonstrated in the field. Initial results show that the stereo system can achieve up to 8Hz effective frame rate with long range obstacle detection and robustness to illumination variations. In order to make the use of stereo in long missions over rough terrain more practical, we have developed techniques for self-calibration of a set of stereo cameras [Jiar and Hebert, 1997].

Second, we have developed a high-performance laser range finder under partial support from a state/industry partnership program. This laser range finder can achieve 100KHz measurement rates with a maximum range of 50m. The sensor provides accurate intensity measurements in addition to range. This sensor will enable faster and more reliable driving in cross-country terrain.

Data from stereo cameras or laser range finder provide data for describing the shape of the terrain. Our third sensing modality is designed for identifying the nature of the terrain. Specifically, we are using a prototype AOTF system developed by the Carnegie Mellon Research Institute (CMRI) under a MURI program. This sensor the imaging of a scene at arbitrary frequency and polarization settings which can be programmed in real-time. For example Figure 4 shows a scene imaged at 530nm and 620nm. Selecting the appropriate filter parameters facilitate terrain classification. We have conducted preliminary experiments and are developing a prototype system to be integrated with the UGV system.

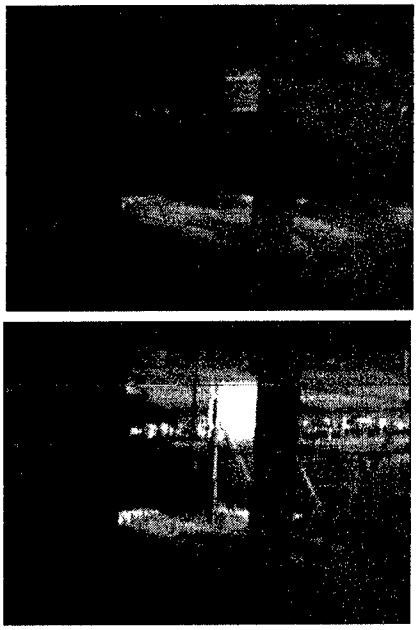


Figure 4: Sensing for terrain typing. A scene imaged at two different wavelengths, 620nm (top) and 530nm (bottom).

In the area of perception for UGV navigation, we have developed an algorithm for representing uncertainty in our obstacle detection system, SMARTY. Specifically, obstacle regions are assigned confidence values based on a sensor model and on the number of observations of the obstacle region. Regions of low confidence are identified as spurious detections and are eliminated

from the map. This approach to perception for navigation was shown to enhance performance by removing false obstacles due to sensor noise, or terrain features such as vegetation. Based on this definition of uncertainty, we have developed an algorithm for sensor control which integrates optimization of terrain coverage and reduced uncertainty in obstacle detection.

In the area of planning, we have extended the capabilities of our two systems, D*, a dynamic route planner, and DAMN (Distributed Architecture for Mobile Navigation), an arbitration system for integrating commands from different mobility components. We have extended D* to the case of multiple agents and multiple goals. The new system, GRAMMPS, is able to dynamically plan route for several vehicle simultaneously, and to dynamically allocate goals [Brumitt and Stentz, 1996]. The map used in both D* and GRAMMPS is updated every time new sensor data becomes available. The extension of our planning system to multiple agents greatly increases the range of UGV missions.

Our current approach to integrating multiple driving behaviors into a system is to use an arbiter to combine the commands issued by the behaviors. This approach has been successfully demonstrated in scenarios in which the vehicle drives at moderate speed. At higher speeds, however, this approach to command arbitration leads to instability because of delays and latencies that are not taken into account in the arbiter. We have developed a new approach which addresses those issues [Rosenblatt, 1996]. In this approach, each mobility behavior contributes a local map in which areas that are driveable and consistent with the task of the behavior are assigned high "utility" values, while areas such as obstacles are assigned low utility values. The arbiter combines the utility maps from all the behaviors into a single map from which the optimal command, speed and turn radius, is computed. We have shown that this approach provides increased stability and robustness at higher speed.

Finally, in the area of teleoperation, we have conducted controlled user studies in order to evaluate our approach to teleoperation, STRIPE [Kay, 1996]. In this approach, the user selects points in images transmitted from the vehicle. The points are transformed to three-dimensional locations as the vehicle travels. STRIPE is unique in that it is designed for operation with very low bandwidth communication links in rough terrain. The purpose of the users studies was to evaluate the users' response to different types of interfaces, and to different system parameters, e.g., system latency, image resolution. The studies provide a solid foundation for further development of teleoperation systems.

4.2. Planetary Rovers

We have made progress in perception for planetary rovers in three major areas: long-duration autonomous navigation, landmark-based position estimation, and augmented reality.

We (Krotkov, Hebert and Simmons) have demonstrated the combining use of stereo and safeguarding laser range finder for autonomous navigation. Specifically, we have demonstrated navigation of over a 50km course in the "Moonyard", a test site designed to simulate lunar terrain [Fuke and Krotkov, 1996, Krotkov et al., 1995, Krotkov et al., 1996]. The navigation system was based on correlation area-based stereo running on a Pentium processor, with frame rates of 2Hz. In addition, a short range laser range finder provided safeguarding for those obstacles not detected by stereo.

Landmark-based position estimation enables rover navigation in the absence of external active positioning reference. We (Deans, Krotkov and Hebert) have started the development of a positioning system based on tracking natural features in a stream of images. This visual feature tracker is being integrated into a system which will use the estimated location of the feature to update the position estimate of a rover which tracking the fixed feature relative to the rover as the it traverses a site

.We (Cozman, Krotkov) have been developing an interface aimed at mobile robot operations in space, which operates by analyzing images sent by the robot and overlaying information about the robot's environment onto the images [Cozman and Krotkov, 1996, 1997]. To this end, we have developed algorithms for pose estimation from outdoor imagery, which allow us to automatically detect the position of the robot and determine the relationship between terrain information and image pixels. We

have processed Earth images from Pittsburgh (Pennsylvania), Dromedary Peak (Utah) and Niles Canyon and Don Juan Resort (both in California), with accuracies ranging from 80 to 150 meters. We have also obtained and mosaiced a sequence of images from the Apollo 17 mission, and applied our algorithms to this sequence with resulting accuracy of 300 meters. The implemented system achieves better estimation performance than competing methods, due to our quantitative approach and better time performance due to our pre-compilation of relevant data.

Terrain information and position estimates are presented to the operator so as to increase situational awareness and prevent loss of orientation. To be able to continuously superimpose relevant information on the incoming sequence of images, we estimate the motion between frames using the Kuglin-Hines algorithm. Motion estimates allow us to stabilize video sequences and create seamless panoramas with broad field of views. The complete system, from real-time image acquisition to map rendering, is now integrated in a single system running in a Silicon Graphics Impact workstation. Image processing and position estimation takes an average of 2 to 3 seconds. The complete system, with image mosaicing, mountain detection and position estimation will be used in the coming Nomad mission, a multi-week traverse of the Atacama desert, Chile, to be conducted by the Robotics Institute at CMU.

4.3. Automated Highway Systems Program

We (Pomerleau, Jochem, Sukthankar, Rosenblatt, Kay, Langer and Thorpe) have been working two sub-project in the DOT work. The first, Run Off Road Collision Countermeasures, is not aimed at autonomous driving, but rather at driver assist. The goal is to have a computer vision system monitor the vehicle's position in the lane while a person drives. Then, if the person starts to fall asleep and drift off the road, the computer can wake the driver before a collision occurs. The first phase of this project is now complete. It consisted of statistical analysis of the accident data to determine the causes of accidents, computer simulations of accident trajectories to identify the opportunities and times for intervention, prototyping of a vision system for determining lane position, and experiments in a driving simulator to measure human reaction to various warning systems. The results of this first phase are encouraging. 80.4% of single vehicle roadway departure accidents are due to driver error, and 64.8% of those could potentially be prevented by warning of too high a speed coming into a corner or drifting out of the lane. RALPH [Pomerleau and jochem, 1996, 1996b], the vision system built for this project, tracks lane positions to within 12 cm even in inclement weather. Test subjects in the University of Iowa driving simulator react well to auditory alarms or haptic alarms, nudges on the steering wheel back towards the road center. The next phase of the project is now under way. This consist of building a new test vehicle, the Navlab 8, and performing on the road tests. The first set of tests will use RALPH in a passive mode, to measure typical lane-tracking behavior of several test drivers on a variety of roads. This will be used to set lane departure warning thresholds low enough to not generate false alarms, but sensitive enough to provide ample warning. The next set of tests will involve extended duration tests of the complete warning system, testing both drivers in the Navlab 8 minivan and professional truckers.

The second DOT project is the Automated Highway System. The goal of the AHS project is to provide completely automated driving of specially-equipped vehicles on specially-equipped lanes. An AHS vehicle will look like an ordinary car, truck, or bus, and will be driven normally until it merges onto the AHS freeway. Then, the driver will designate a destination, and the automated system will take control and drive smoothly and safely until the desired exit is reached. AHS is being developed in the US by a consortium, which includes General Motors, Delco, Hughes, Bechtel, Parsons Brinkerhoff, the University of California, Caltrans, and Lockheed Martin, and the US DOT, as well as

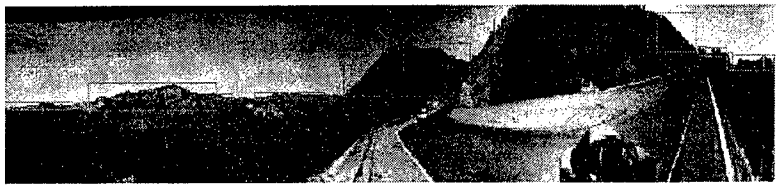


Figure 5: Result of mosaicing a sequence of 20 images while detecting the skyline and the mountains.

CMU. The project is proceeding along several parallel fronts. In technology development, the main questions involve how to follow the road, and how to detect obstacles. CMU is building a new ladar, a new radar, enhancements to the RALPH vision system, stereo vision, carrier phase GPS for blind driving, and optical flow for obstacle detection. In concept development, one of the major issues is whether the automated vehicles need to drive in a separate lane, occupied only by other automated vehicles, or whether they can be made intelligent enough to drive intermingled with human-driven cars. CMU is building systems to recognize other vehicles in traffic, predict their behavior, and maneuver safely through the traffic stream. As part of the legislation authorizing the AHS project, Congress asked for a proof of technical feasibility demonstration in 1997. The integrated demo will take place the first week of August on the HOV lanes of I-15, just north of San Diego CA. CMU is building 5 new vehicles for the 97 demo: Navlabs 6 and 7 (two Pontiac Bonnevilles, partially instrumented by General Motors and Delco), Navlab 8 (an Oldsmobile Silhouette minivan), and Navlabs 9 and 10 (two Flexible city busses). The 5 vehicles will demonstrate lane following, lane changing, vehicle following, vehicle passing, and obstacle detection and avoidance.

5. Image Understanding for Intelligent Video and Image Databases

Digital image and video databases are rapidly becoming important for education, entertainment, and a host of multimedia applications. Because of the size of the image and video collections, technology is needed to effectively store, browse, retrieve, and present the content that a user wishes to access.

5.1. Video Skimming

The Informedia™ Digital Video Library at Carnegie Mellon University, funded by NSF, ARPA, and NASA, is developing intelligent, automatic mechanisms to populate a video library and to allow for full-content knowledge-based search, retrieval and presentation. The distinguishing feature of Informedia's approach is the integrated application of speech, language and image understanding technologies for efficient creation and exploration of the library.

We (Smith and Kanade) have been developing a method to extract the significant audio and video information and create a "skim" video - a very short synopsis consisting of the significant words and images - which enables the user to quickly grasp the entire content of the original video [Smith and Kanade, 1997]. The goal is to automatically reduce the playback time by as much as 20:1, e.g., to compact an hour-long video to a clip as short as a few minutes.

The current method consists of language analysis and image analysis, plus skimming rules. The image analysis part segments the input video into scenes by detecting scene breaks. Camera motions are detected and classified into pan, left/right, zoom in/out, and partial motion (usually, object move). Also, important objects are detected. At present, the methods enables detection of faces [Rowley, Baluja and Kanade, 1997] and text.

The language analysis part extracts relevant key words and phrases by using the well-known technique of Term Frequency Inverse Document Frequency (TF-IDF). The TF-IDF of a word is its frequency in a given scene, f_s, divided by the frequency, f_c, of its appearance in a standard corpus. Words that appear often in a particular segment, but relatively infrequently in a standard corpus, receive the highest TF-IDF weights.

Finally, the skim rules combine the results of the two analyses. The audio part of the skim video consists of a sequence of phrases or sentences that include the key words. The video part, however, is not simply a concatenation of the corresponding video, since the video and audio parts of videos are not usually synchronized. Instead, the video parts are determined by rules such as, "If a key word is human related, select the nearby video portion that includes faces" and "If video consists of panning followed by static and then zoom-in, the last part is usually the most important."

We have created several skim videos by using this system; their compaction ratio ranges from 6 to 15. Currently, user study is being performed to evaluate the effectiveness of video skimming.

5.2. Neurosurgical Database

We (Liu, Chen and Kanade) are developing an intelligent, interactive, Content-based Medical Image Storage and Retrieval (CMISR) System as part of the National Medical Knowledge Bank project. Medical images form an essential and inseparable component through out the diagnosis and treatment process. This is especially true in neurology, which is our current research focus. The CMISR system uses medical images and their collateral information, such as relevant medical records and procedures in the form of text, voice or video, to achieve the three main goals of the knowledge bank: 1) facilitating consultations between primary care givers and specialists,; 2) improving patient management, and 3) enhancing medical education and training.

The specific function of the Knowledge Bank is to find medical cases similar to a patient's case at hand. Accordingly, the CMISR system will ultimately have the ability for *similarity-based medical image storage and retrieval.* For indexing neuroradiology images, with the help from radiologists and neural surgeons, we have identified, and are in the process of building detectors for a set of salient visual features, including: mass effect, anatomical location, density, sensitivity to contrast enhancement, edema, shape and boundary.

6. Image-Guided Computer-Assisted Surgery

Surgical precision is of vital importance in many fields such as neurosurgery and orthopaedics. Computer-assisted surgical systems can greatly increase the accuracy of these operations and have already been applied to a few applications.

Real-time and interactive imaging of complex biomedical systems has become another priority within medicine. One major challenge is to integrate the precise information currently found with CT and MRI into surgical practice.

6.1. Accurate Shape-based Registration

The registration process is a fundamental component of most computer-assisted surgical systems. Registration estimates a spatial transformation between two coordinate systems: a pre-operative system used to construct plans or simulations based upon medical data (e.g., CT, MRI, or X-ray images), and an intra-operative system in which the surgical procedure is performed (e.g., relative to a robot, navigational guidance system, etc.).

We (Simon and Kanade) have developed registration methods, referred to as *shape-based registration* methods, which use representations of object shape to estimate the required transformation. Representations are constructed using data collected in the two coordinate systems (i.e., pre- and intra-operative). Registration estimates a transformation which aligns one shape representation with the other in a manner which minimizes a measure of the distance between them.

Several factors affect shape-based registration accuracy, including: errors in the shape representations due to sensor noise or shape reconstruction errors [Simon et al., 1995]; the quantity of registration data; and the locations on the registration object from which the data are collected [Simon et al., 1995, Simon, 1996]. This work addresses the problem of improving shape-based registration accuracy via intelligent selection of registration data and on-line estimation of accuracy. Intelligent data selection (IDS) is comprised of geometric constraint analysis which provides a sensitivity measure shown to be well correlated with registration accuracy; and geometric constraint synthesis, an optimization process which generates data configurations which maximize the sensitivity measure for a fixed quantity of data. IDS uses the pre-operative shape representation to generate a data collection plan (DCP) which can be used during surgery to guide the acquisition of registration data. On-line accuracy estimation provides an upper bound on true registration accuracy based upon a conventional root-mean-squared error.

After in-vitro on cadaveric specimens and via simulation studies, the above method has been incorporated into the HipNav system, a clinical image-guided orthopedic surgical application [Simon et al., 1997] [Simon and Kanade, 1997] [Jaramaz et al., 1997].

6.2. 3-Dimensional Image Overlay

Image overlay is a display technique that combines 2D or 3D computer generated images with the user's view of the real world. We (Blackwell, Morgan, Simon and Kanade) have developed a system that provides the observer with an unimpeded view of the actual environment, enhanced with 3D stereo images [Mrcas, 1997]. The system tracks objects in the real world and the observer's view point to transform the computer images to appear in the appropriate location (see Figure 6)

A significant advantage of our 3D image overlay system is that the user sees virtual images properly registered within the real world scene. For some tasks, the ability to view this data without looking away from the scene is extremely beneficial. The user views the patient through a beam splitter (a half-silvered mirror) which is both transparent and reflective. Positioned above the beam-splitter is a display device (video monitor or projector). The user sees the patient directly through the beam-splitter, and also sees a reflection of the video display which appears to float within the workspace. A pair of liquid crystal shutter glasses are worn allowing the user to view stereo images. A 6 degree of freedom tracking system is integrated with the overlay device allowing the user to change the view point and objects in the world scene to move while maintaining correct registration of the images.

Figure 6: Stereo image

The current prototype system is being re-designed as a smaller and more mobile system for use in an operating room.The system is also being designed for use in Interventional Magnetic Resonance Imaging (IMRI) systems, which will allow a surgeon to view MRI images in near real-time with a simultaneous unobstructed view of the patient.

References

[Baluja et al., 1996] S. Baluja, R. Sukthankar, and J. Hancock, "Prototyping Intelligent Vehicle Modules Using Evolutionary Algorithms." To appear in Evolutionary Algorithms in Engineering Applications, Dasgupta, D. and Z. Michalewicz, eds., pub. by Springer-Verlag, 1996.

[Brajovic and Kanade, 1994] V. Brajovic and T. Kanade, "Computational Sensors for Global Operations", *Proc of IUW*, pp. 621-630, 1994.

[Brajovic and Kanade, 1996] Brajovic, V. and T. Kanade, "A Sorting Image Sensor: An Example of Massively Parallel Intensity-to-Time Processing for Low-Latency Computational Sensors," *Proc. of IEEE Intern. Conf. on Robotics and Automation*, April 1996, Minneapolis, MN.

[Brajovic and Kanade, 1997] V. Brajovic and T. Kanade, "CMU MURI: Ingegrated Vision and Sensing for Human Sensory Augmentation," in these proceedings.

[Brumitt and Stenz 1996] B. Brumitt and A. Stentz, "Dynamic mission planning for multiple mobile robots,". *Proc. of IEEE Intern. Conf. on Robotics and Automation*. Minneapolis. April 1996.

[Cmisr, 1997] http://www.cs.cmu.edu/afs/cs.cmu.edu/user/yanxi/www/images/medical_image.html.

[Cozman and Krotkov, 1996] F. G. Cozman and E. Krotkov, "Position estimation from outdoor visual landmarks for teleoperation of lunar rovers," *Proc. of Third IEEE Workshop on Applications of Computer Vision*, December 1996.

[Cozman and Krotkov, 1997] F. G. Cozman and E. Krotkov, "Automatic Mountain Detection and Pose Estimation for Teleoperation of Lunar Rovers," *Proc. of IEEE Intern. Conf. on Robotics and Automation*, April 1997.

[Fuke and Krotkov, 1996] Y. Fuke and E. Krotkov, "Dead Reckoning for a Lunar Rover on Uneven Ter-

rain," *Proc. of IEEE Intern.. Conf. Robotics and Automation.* Minneapolis. April 1996, pp. 411-416.

[Hebert, 1996] M. Hebert. Mobile Robots In Unstructured Environments. *Proc. Fourth ICARV.* Singapore. December 1996.

[Hebert et al., 1996] M. Hebert, C. Thorpe, A. Stentz. *Intelligent Unmanned Ground Vehicles.* Kluwer Academic Publishers. 1996.

[Ikeuchi et al., 1996] K.Ikeuchi, T. Shakunaga, M.Wheeler and T.Yamazaki. Invariant Histograms and Deformable Template Matching for SAR Target Recognition. *Proc. IEEE Computer Vision and Pattern Recognition.* San Francisco, CA, June 1996, pp.100-105.

[Jaramaz et al., 1997] B. Jaramaz, "Range of Motion After Total Hip Arthoplasty: Experimental Verification of the Analytical Simulator," *Proc. 1st Joint CVRMed / MRCAS Conf,* Grenoble, March 1997.

[Jiar and Hebert, 1997] Y. Jiar and M. Hebert. *Practical Self-Calibration of Stereo Cameras.* Technical Report CMU-RI-TR-97. The Robotics Institute. Carnegie Mellon University. 1997.

[Johnson and Hebert, 1996] A. Johnson and M. Hebert. *Recognizing Objects by Matching Oriented Points.* Carnegie Mellon Robotics Institute Technical Report CMU-RI-TR-96-04, April 1996. To appear in *Proc. IEEE Conference on Computer Vision and Pattern Recognition.*

[Johnson and Hebert, 1997] A. Johnson and M. Hebert. Surface Registration by Matching Oriented Points. *Proc. International Conference on Recent Advances in 3-D Digital Imaging and Modeling (3DIM '97),* May 1997.

[Johnson et al., 1997b]A. Johnson, R. Hoffman, J. Osborn and M. Hebert. A System for Semi-Automatic Modeling of Complex Environments. *Proc. International Conference on Recent Advances in 3-D DIgital Imaging and Modeling (3DIM '97),* May 1997.

[Kanade and Bajcsy, 1993] T. Kanade and R. Bajcsy, "Computational Sensors: A Report from DARPA workshop", *Proc. of IUW,* 1993.

[Kay, 1996] J. Kay. *Remote Driving with Limited Image Data.* Ph.D. Dissertation. The Robotics Institute. Carnegie Mellon University. November 1996.

[Krotkov et al., 1995] E. Krotkov, M. Hebert, and R. Simmons, "Stereo Perception and Dead Reckoning for a Prototype Lunar Rover," *Autonomous Robots,* Vol. 2, No. 4. December 1995.

[Krotkov et al. 1996] E. Krotkov, R. Simmons, F. Cozman, S. Koenig, "Safeguarded Teleoperation for Lunar Rovers: From Human Factors to Field Trials," *Proc. IEEE Workshop Planetary Rover Technology and Systems.* Minneapolis. April 1996, pp. 411-416.

[Langer, 1977] Langer, D., *An Integrated MMW Radar System for Outdoor Navigation,* PhD Thesis, Carnegie Mellon Robotics Institute, January 1997.

[Liu et al., 1996] Liu, Y. et al., Automatic Extraction of the Central Symmetry (Mid-Sagital) Plan from Neuroradiology Images. Carnegie Mellon University Robotics Institute Technical Report CMU-RI-TR-96-40.

[Mrcas, 1997] http://www.cs.cmu.edu/afs/cs/project/mrcas/www/mrcas-home/overlay.html

[Ohba and Ikeuchi, 1997] K.Ohba and K.Ikeuchi. Detectability, *Uniqueness, and Reliability of Eigen-Windows for Stable Verification of Partially Occluded Objects.* Accepted for publication, IEEE Trans. on Pattern Analysis and Machine Intelligence.

[Pape et al., 1996] D. Pape, D., V. Narendran, M. Koenig, J. Hadden, J. Everson, and D. Pomerleau, *Dynamic vehicle simulation to evaluate countermeasure systems for run-off-road crashes.* SAE Technical Paper 960517, Detroit, Michigan, February 26-29, 1996.

[Pomerleau and Jochem, 1996] D. Pomerleau and T. Jochem, "Life in the Fast Lane: The Evolution of an Adaptive Vehicle Control System," *AI Magazine,* Vol. 17, No. 2 pp. 11-50.

[Pomerleau and Jochem, 1996b] D. Pomerleau and T. Jochem, "Rapidly Adapting Machine Vision for Automated Vehicle Steering," *IEEE Expert,* Vol. 11, No. 2 pp. 19-27.

[Rowley, Baluja and Kanade, 1996] H. Rowley, S. Baluja, and T. Kanade (1996): "Neural Network-Based Face Detection," *Proc. at Int'l Conference on Computer Vision and Pattern Recognition 96 (CVPR '96),* San Francisco, CA, June 18 - 20, 1996, pp.203-208

[Tijeria et al., 1996] L. Tijerina, J. Jackson, D. Pomerleau, R. Romano, and A. Petersen, "Driving Simulator Tests of Lane Departure Collision Avoidance Systems," *Proc. of ITS America sixth Annual Meeting,* Houston, TX.

[Robert et al., 1996] L. Robert, C. Zeller, O. Faugeras, M. Hebert, "Applications of NonMetric Vision To Some Visually Guided Robotics Tasks," *Visual Navigation.* Ed. Yiannis Aloimonos. LEA Publishers. 1996.

[Rosenblatt, 1996] J. Rosenblatt. *A Distributed Approach to Mobile Navigation.* Ph.D. Dissertation. The Robotics Institute. Carnegie Mellon University. September 1996.

[Sato and Ikeuchi, 1994] Y. Sato and K. Ikeuchi, "Temporal-Color Space Analysis of Reflection," J. of Optical

Society of America A, Vol. 11, No. 11, pp. 2990-3002, November

[Sato and Ikeuchi, 1996] Y. Sato and K. Ikeuchi, "Reflectance analysis for 3D computer graphics model generation," *Graphical Models and Image Processing,* Vol. 58, No. 5, pp. 437-451, September, 1996

[Sato, 1997] Y. Sato, *Object Shape and Reflectance Modeling from Color Image Sequence,* Ph.D dissertation,Technical Report CMU-RI-97-06, The Robotics Institute, Carnegie Mellon University, February 1997.

[Simon, 1995] D. A. Simon, "Accuracy validation in image-guided orthopaedic surgery," *Proc. 2nd Intern. Symp. MRCAS,* Baltimore, Nov. 1995.

[Simon, 1996] D. A. Simon, *Fast and Accurate Shape-Based Registration* Ph.D dissertation, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, December 1996.

[Simon et al., 1995] D. A. Simon, M. Hebert, and T. Kanade, "Techniques for fast and accurate intra-surgical registration," *Journal of Image Guided Surgery,* Vo. 1, No.1, pp.17-29, April 1995.

[Simon et al., 1997] Simon, D.A., et al., "Development and validation of a navigational guidance system for acetabular implant placement," *Proc. 1st Joint CVRMed / MRCAS Conf.,* Grenoble, March 1997.

[Simon and Kanade, 1997] D. A. Simon and T. Kanade, "Geometric Constraint Analysis and Synthesis: Methods for Improving Shape-based Registration Accuracy," *Proc. 1st Joint CVRMed/MRCAS Conf.,* Grenoble, March 1977.

[Smith and Kanade, 1997] M. A. Smith and T. Kanade, "Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques," Technical Report CMU-CS-97-111, Carnegie Mellon University, February 3, 1997.

[Sukthankar et al., 1996] R. Sukthankar, J. Hancock, S. Baluja, D. Pomerleau, and C. Thorpe, "Adaptive Intelligent Vehicle Modules for Tactical Driving," *Proc. AAAI Workshop on Adaptive Intelligent Agents,* Seattle, Washington.

[Sukthankar et al., 1996a] R. Sukthankar, J. Hancock, D. Pomerleau, and C. Thorpe, "A Simulation and Design System for Tactical Driving Algorithms," *Proc of AI, Simulation and Planning in High Autonomy Systems (AISP '96),* 1996.

[Suktanker et al., 1996b] R. Sukthankar, J. Hancock, and C. Thorpe, "Tactical-level Simulation for Intelligent Transportation Systems." To appear in *Journal on Mathematical and Computer Modeling,* Special Issue on ITS, 1996.

[Sukutanker, 1997] Sukthankar, R. *Situational Awareness for Driving in Traffic,* Ph.D Dissertation, Carnegie Mellon Robotics Institute, January 1997.

[Takeuchi, 1997] Y. Takeuchi, P. Gros, M. Hebert, K. Ikeuchi, *Visual Learning for Landmark Recognition,* Technical Report CMU-RI-TR-97. The Robotics Institute. Carnegie Mellon University. 1997.

[Thorpe and Pomerleau, 1996] C. Thorpe and D. Pomerleau, "Robots, Transportation, and Society," *Proc of Intern. Workshop on Advanced Robotics and Intelligent Machines,* University of Salford, Manchester, U.K., April 1996

[Thorpe and Hebert, 1996] C. Thorpe and M. Hebert, "Mobile Robotics: Perspectives and Realities," *Proc. of Intern. Workshop on Advanced Robotics and Intelligent Machines,* University of Salford, Manchester, U.K., April 1996.

[Wheeler and Ikeuchi, 1993] M. D. Wheeler and K. Ikeuchi, "Sensor Modeling, Probablistic Hypothesis Generation, and Robust Localization," *IEEE Trans Pattern Analysis and Machine Intelligence,* Vol. 17, No. 3, pp. 252-265, March.

[Wheeler, 1996] M.D. Wheeler. *Automatic Modeling and Localization for Object Recognition.* Ph. Dissertation. Technical Report CMU-CS-96-188. Computer Science Department. Carnegie Mellon University. 1996.

[Zhang, 1996] D. Zhang, M. Hebert. Scale space classification of 3D objects. *Technical Report CMU-RI-TR-96-236.* The Robotics Institute, Carnegie Mellon University. To appear in *Proc. IEEE Conference on Computer Vision and Pattern Recognition.*

VIDEO SURVEILLANCE AND MONITORING (VSAM) TECHNICAL PAPERS

Exploring Visual Motion Using Projections of Motion Fields

Sándor Fejes and Larry S. Davis
Center for Automation Research
University of Maryland
College Park, MD 20742-3275

Abstract

The dimensionality of visual motion analysis can be reduced by analyzing projections of flow vector fields. In contrast to motion vector fields, these projections exhibit simple geometric properties which are invariant to the scene structure and depend only on the camera motion. Using these properties, structure and motion can be either completely or partially decoupled. We estimate motion parameters from projections of flow fields by using robust techniques, implemented in a recursive observer model. The recovered scene structure is collected in a visual scene memory for both temporal and spatial integration of instantaneous measurements. The model is applicable to general camera motion and to almost arbitrary known focal length without the requirement of point correspondence. We demonstrate our projection method on the problem of detecting independently moving objects from a moving camera. Using the projection approach, the problem can be reduced to a one-dimensional optimization process which involves robust line-fitting and outlier detection.

1 Introduction

Problems in visual motion analysis require the study of large, relatively high-dimensional spatio-temporal data (visual displacement vector fields) projected onto the image plane. Each component of this data provides only a weak geometric constraint on the confounding of camera motion and scene structure, and the input data is usually highly erroneous. Therefore, visual-motion-based problems such as navigation or structure estimation need to be addressed by a combination of geometry and robust spatio-temporal signal processing.

In our approach to analyzing visual motion fields, we propose a method based on the analysis of projected components of flow vector fields. When the flow vector field is projected onto a small set of lines, the original two-dimensional flow field is reduced to a small number of one-dimensional scalar functions representing components of the input flow field. Exploiting two simple properties of the projected flow fields allows us to recover the projected components of the egomotion. To obtain complete information about the motion two (or more) projections in different (e.g. orthogonal) directions can be employed. In certain restricted domains, even the partial information provided by a single projection is sufficient to obtain information about the motion and the scene structure. Emphasizing the importance of integral measurements, we extend our analysis of flow fields into the spatio-temporal domain by using a cumulative visual scene memory.

In the first part of this paper we introduce our imaging model and characterize visual motion fields. We then examine the projected components of these fields and discuss various properties of special one-parameter families of these projections. Based on these properties we construct an algorithm and implement it in the form of a recursive filter for the estimation of egomotion parameters. We demonstrate the application of the projection approach to a special case of the structure-from-motion problem, the detection of independently moving objects.

2 Visual motion fields

An important goal of computer vision is to obtain information about (1) the motion of the visual observer and/or (2) the structure of the scene being observed. The camera collects information about the three-dimensional world by projecting it onto a frontal image plane. We define our reference coordinate system in the usual way, as fixed to the camera frame with origin coinciding with the lens center, xy-plane parallel to the image plane, and Z-axis in-

tersecting the image plane at distance f.

Setting the focal length f to unity, the visual motion of an image point $\boldsymbol{q}(x, y, f)$ can be expressed as a function of the camera motion by the well-known motion equation [6]

$$u(x,y) = \overbrace{-\frac{1}{Z_0(x,y)}(x - x_0)}^{u_T} + \overbrace{\Omega_x xy - \Omega_y(1 + x^2) + \Omega_z y}^{u_\Omega} \quad (1)$$

$$v(x,y) = \overbrace{-\frac{1}{Z_0(x,y)}(y - y_0)}^{v_T} + \overbrace{\Omega_x(1 + y^2) - \Omega_y xy - \Omega_z x}^{v_\Omega} \quad (2)$$

where $(x_0, y_0) \stackrel{def}{=} (\frac{U}{W}, \frac{V}{W})$ is the focus of expansion (FOE), $\boldsymbol{T} = (U, V, W)$ denotes the translational velocity and $\boldsymbol{\Omega} = (\Omega_x, \Omega_y, \Omega_z)$ the angular velocity of the camera. $Z_0 \stackrel{def}{=} \frac{Z}{W}$ defines the scaled depth.

Given the flow field $\boldsymbol{v} = (u, v)$, motion analysis aims to recover either the egomotion parameters or the (scaled) scene depth, or both. This problem requires the decoupling of structure from motion by solving (1) and (2) at each flow vector $\boldsymbol{v} = (u, v)$. In general the solution can be very ambiguous, making the problem unstable or ill-posed.

A possible approach to attacking this ill-posed problem is to avoid the complete estimation of the motion parameters and to recover only a reduced, well-conditioned subset of them. This can be done (e.g.) by using parallax [7] or epipolar constraints [13]. Unfortunately, these models require point correspondences, which are hard to extract accurately. Alternative approaches involve the direct use of the so-called normal flow. This form of input data is easy to estimate, providing a large number of constraints on the motion. However, the constraint provided by each normal flow vector is rather weak, which causes high algorithm complexity and instability of the implementations.

3 Projected components of visual motion fields

It is well known that purely translational ($\boldsymbol{\Omega} = \vec{0}$) or purely rotational ($\boldsymbol{T} = \vec{0}$) flow fields possess simple geometric regularities; however, these regularities disappear if the two components are superimposed. This is, evidently, due to the relatively high dimensionality of the resulting flow field.

One way to reduce dimensionality is to use projections. One can consider only specific components of the flow field (in one or more directions) and use them to extract information about the visual motion. This approach was first introduced in [3; 4; 9] and then further developed in [2; 10; 12]. Let $\boldsymbol{v}(x, y)$ be a flow vector field in the image plane and $\boldsymbol{p}$ be the (unit) *projection vector*. Taking the component of $\boldsymbol{v}(x, y)$ parallel to $\boldsymbol{p}$, we obtain a two-dimensional (scalar) function $u'(x, y) = \boldsymbol{v}(x, y)\boldsymbol{p}$.

In order to completely exploit the original flow field we have to perform two or more projections in different (e.g. orthogonal) directions. The fact that no correspondence is established between the components u'_q and u''_q of different projections of the same flow vector $\boldsymbol{v}_q = (u_q, v_q)$ is an important benefit of the method for vision applications: it allows the approach to handle the well-known aperture problem by not requiring point correspondence.

In the special case when $\boldsymbol{p}$ is parallel to the x-axis we obtain the identity $u'(x, y) = u(x, y)$ of (1). Generally, the expression for u' remains of the same form due to symmetry considerations, except that its parameters are expressed in the rotated coordinate system aligned with $\boldsymbol{p}$. In practice it may be that some projection directions are better than others, as we will see; however, this will not effect the theoretical discussions to follow. Therefore, for simplicity we will consider the x-component only in the standard image coordinate system (x, y) and refer to other coordinate systems (x', y') only when necessary.

In our analysis of projections of flow fields we will consider *restrictions* on $u(x, y)$ which will result in one-parameter-functions of u. We define the *parallel restriction*

$$u_y(x) \stackrel{def}{=} u(x, y)\Big|_{y=const}, \quad (3)$$

where the restriction (sampling) direction is *parallel* to the projection direction $\boldsymbol{p}$, and the *orthogonal restriction*

$$u_x(y) \stackrel{def}{=} u(x, y)\Big|_{x=const}, \quad (4)$$

where the restriction (sampling) direction is *orthogonal* to the projection direction $\boldsymbol{p}$ (Figure 1). We show in the next two subsections that these two restrictions reveal qualitatively different properties of the underlying motion field.

3.1 Parallel restriction of projected flow

The parallel restriction on a projection of a flow field can be decomposed into translational and rotational components (1): $u_y(x) = u_{y,T}(x) + u_{y,\Omega}(x)$.

The translational component is a function of scene depth and the distance from the projection of the

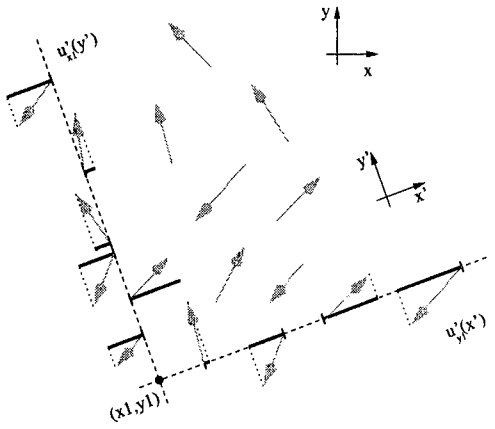


Figure 1: One-parameter functions of the projections of flow fields: the parallel restriction u_y and the orthogonal restriction u_x.

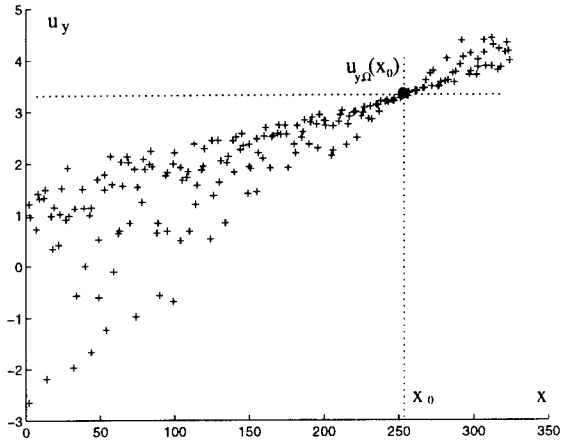


Figure 2: The divergence property of the parallel restriction of projected flow in the case of general camera motion and narrow FOV.

FOE, x_0. For the point $x = x_0$ the translational component vanishes, independently of the underlying structure.

The rotational component $u_\Omega(x)$ is in general a second-order function of x. Assuming that the image FOV is small ($x, y \ll 1$), the second-order terms of x and y can be neglected, so that we have $u_y(x) = \frac{x-x_0}{Z_0(x,y)} - \Omega_y + \Omega_z y$. Since the rotational terms have become constant with respect to x we can make the following observation:

$$\forall x : \begin{cases} x \leq x_0 & \Rightarrow u_y(x) \leq u_y(x_0) = u_{y,\Omega}(x_0) \\ x \geq x_0 & \Rightarrow u_y(x) \geq u_y(x_0) = u_{y,\Omega}(x_0). \end{cases} \quad (5)$$

These inequalities express the **divergence property** of parallel restrictions (Figure 2). They reflect the assumption that scene depth must be positive (because the scene is in front of the camera). The divergence property of purely translational flow fields is widely known. Less obvious is the regular behavior of general flow fields if a specific projection is considered. Of course, as the image FOV becomes large the rotational component depends on x, which cannot be ignored, and the divergence property no longer holds.

In summary, the parallel restriction provides constraints on the location of the FOE if the FOV is small, or more generally if the rotational component is negligible compared to the translational one.

3.2 Orthogonal restriction of projected flow

Like the parallel restriction, the orthogonal restriction can also be decomposed into translational and rotational components: $u_x(y) = u_{x,T}(y) + u_{x,\Omega}(y)$.

From (1) it follows that when $x = x_0$ the translational component $u_{x_0,T}(y)$ vanishes for all y independently of the structure of the scene.

The rotational component of the orthogonal restriction is a linear function of the free variable, y, which is independent of the camera parameters. In the case of a narrow FOV, it reduces to $u_{x,\Omega}(y) = -\Omega_y + \Omega_z y$. We will call this the **linearity property** of the orthogonal restriction. An orthogonal restriction, when it intersects the FOE, is a linear function of y; therefore the linearity property provides a constraint on the rotational parameters.

It is important to note that the linearity property is not a sufficient condition for concluding that a particular restriction intersects the FOE. If (e.g.) the flow field contains only rotational components (e.g., $Z_0 = \infty$), or the scene structure is *critical* ($\frac{1}{Z_0(x,y)} = r_0(x) + r_1(x)y$), then the orthogonal restriction is linear whether or not it intersects the FOE. It can be easily shown that one of the most common scene structures, a plane, is also a *critical surface* for orthogonal restrictions; planes induce linear orthogonal restriction functions independently of camera motion.

4 Recovering motion using projections of flow fields

The divergence and linearity properties of restrictions of projected flow fields provide constraints on the camera motion and can be used as a basis for estimating motion parameters.

In [2; 3; 12] the linearity property of projections of flow fields is used to constrain the location of the FOE and estimate the rotational parameters. In those approaches, the parameter estimation search process analyzes only those orthogonal projections which pass through the image center ($x = 0$). This

reduces (1) to the very simple form $u_{x=0}(y) = -\Omega_y + \Omega_z y$, independent of the camera parameters. This constraint is used to find the direction of the line passing through the FOE. A second search along this line determines the location of the FOE; one can then immediately recover the rotational components.

While this method is very elegant and is independent of the camera parameters, it does suffer from several shortcomings. First, the approach uses only the flow values that fall on two intersecting lines as the basis for motion parameter estimation. Second, as we have seen, the linearity property is only a necessary and not a sufficient condition for locating the FOE.

In what follows we describe an algorithm that combines the linearity and the divergence properties to solve motion estimation problems, but does so using methods that integrate information over the entire image plane.

4.1 The algorithm

First, for simplicity, we will assume that the camera FOV is narrow and that the projection of the FOE (x_0) is inside the image. Choosing projection direction $\boldsymbol{p}$,

1. We analyze each parallel restriction, $u_y(x)$, and for each x find the value $\hat{u}_{y,\Omega}(x)$ that best satisfies the divergence property under the assumption that $x = x_0$. This corresponds to the value that minimizes the regions of negative depth [5] in the parallel restriction. Geometrically, it is that value $\hat{u}$ such that the NW and SE quadrants of the $x - u$ plane defined by the axis through $(x, \hat{u})$ have (e.g.) the minimum numbers of projected flow values (Figure 3). A straightforward algorithm for computing $\hat{u}_{y,\Omega}$ would require $\mathcal{O}(N^2)$ computation for $\mathcal{O}(N)$ steps of x and u. In Section 4.2 we present a heuristic algorithm for computing $\hat{u}_{y,\Omega}(x)$ in $\mathcal{O}(N)$ time.

2. These $\hat{u}_{y,\Omega}(x)$ are combined into $\hat{u}_\Omega(x, y)$ and the orthogonal restrictions $\hat{u}_{x,\Omega}(y)$ are constructed. The linearity property is then used to select the estimate $\hat{x}_0$. According to the linearity property $u_{x_0}(y) = -\Omega_y + \Omega_z y$; therefore we select the x value for which the orthogonal restriction $\hat{u}_{x,\Omega}(y)$ best fits a line at $\hat{x}_0$. Furthermore, $\hat{\Omega}_y = \text{mean}\{\hat{u}_{\hat{x}_0,\Omega}(y)\}$ and $\hat{\Omega}_z = \text{slope}\{\hat{u}_{\hat{x}_0,\Omega}(y)\}$

4.2 Implementation

We describe a linear algorithm to approximate $\hat{u}_{y,\Omega}(x)$ by defining the *lower* and *upper distributions*

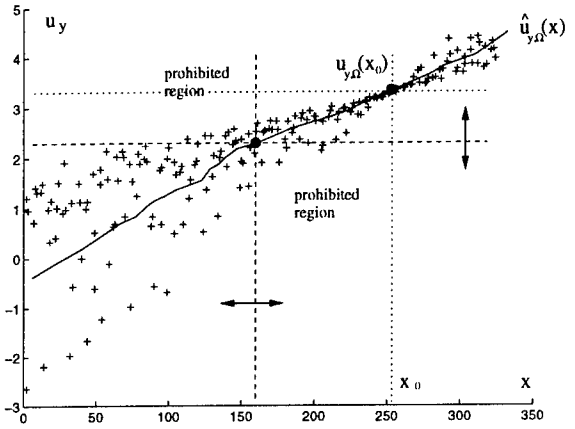


Figure 3: The estimation of $\hat{u}_{y,\Omega}(x)$ in parallel restriction u_y.

of a function u for a given interval $[x_{\min}, x_{\max}]$:

$$u^L(x) \stackrel{def}{=} \begin{cases} \min_{\xi \geq x}\{u(\xi)\} & \text{if } x \leq x_{\max} \\ +\infty & \text{otherwise,} \end{cases} \quad (6)$$

$$u^U(x) \stackrel{def}{=} \begin{cases} \max_{\xi \leq x}\{u(\xi)\} & \text{if } x \geq x_{\max} \\ -\infty & \text{otherwise.} \end{cases} \quad (7)$$

Since we have assumed a small FOV and an FOE in the image, the parallel restriction satisfies the divergence property. We therefore have the constraint

$$x_0 \in \{x \mid u_y^L(x) \geq u_y^U(x)\}.$$

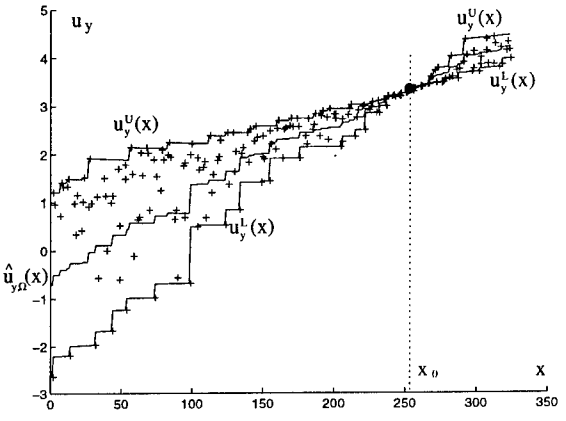


Figure 4: The estimation of $\hat{u}_{y,\Omega}(x)$ in parallel restriction $u_y(x)$ using the lower and upper distributions $u_y^L(x)$ and $u_y^U(x)$.

We use these distributions to, heuristically, estimate $\hat{u}_{y,\Omega}(x)$ as follows (Figure 4):

$$\hat{u}_{y,\Omega}(x) \stackrel{def}{=} \frac{1}{2}\left(u_y^L(x) + u_y^U(x)\right). \quad (8)$$

This is based on the observation that the only possible values for $\hat{u}_{y,\Omega}(x)$ lie in the interval determined

by $u_y^L(x)$ and $u_y^U(x)$, and on the assumption that the flow values will be uniformly distributed in the (x, u) plane between these distribution functions.

The second device for simplifying the computation of $\hat{u}_{y,\Omega}$ is based on reducing the number of parallel restrictions by collapsing a *band* of parallel restrictions of width $2w$ using their extrema as follows:

$$u_k^{\min}(x) = \min_{(2k-1)w \le y < (2k+1)w} \{u_y(x)\}$$
$$u_k^{\max}(x) = \max_{(2k-1)w \le y < (2k+1)w} \{u_y(x)\},$$

where $\frac{y_{\min}}{2w} \le k \le \frac{y_{\max}}{2w}$, and w is the half-width of the band. As a result, the image is subdivided into a small number of non-overlapping bands, where the original parallel restrictions are replaced by the upper and lower envelopes of $u(x, y)$ within each band[1]. This step can be regarded as an extension of the projection principle.

After the collapsing step, the envelopes $u_k^{\min}$ and $u_k^{\max}$ are used to compute $u_k^L(x)$ and $u_k^U(x)$, respectively, and then (8) is used to compute $\hat{u}_{k,\Omega}$. As a result of the collapsing of the image into bands, the line fitting has only $\frac{y_{\max}-y_{\min}}{2w}$ points to estimate the best fit $\overline{u}_{k,\Omega}(x)$ to the values of $\hat{u}_{k,\Omega}(x)$ at each x position (Figure 5). The candidate for x_0 is that x which minimizes the cost function

$$C(x) = \sum_k c_1 \mathrm{pos}^2(\overline{u}_{k,\Omega}(x) - \hat{u}^L_{k,\Omega}(x)) + c_1 \mathrm{pos}^2(\hat{u}^U_{k,\Omega}(x) - \overline{u}_{k,\Omega}(x)) + c_2 \hat{u}^2_{k,\Omega}(x), \quad (9)$$

where the function $\mathrm{pos}(x) = x$ if $x > 0$, else 0, and c_1, c_2 are constants. The first two terms penalize negative depth in the estimated $\overline{u}_{k,\Omega}(x)$, and the last term provides regularization.

Performing these steps, we recover the projected components of the motion. In order to recover the complete set of motion parameters we must employ more (≥ 2) projection directions and combine the results as vector components.

4.3 Extension of the algorithm to a general FOV

Until now we have assumed that the FOV is small, since the divergence property is not satisfied for a large FOV when the flow contains a large rotational component. The extension of the algorithm to the case of a wide FOV is as follows. We process the flow field as if the FOV were narrow. We use the estimated rotational parameters to derotate the original flow, and we iteratively process the derotated flow, in which the rotational component has been reduced. As the iteration process converges, the rotational component decreases and the divergence property is better satisfied by the derotated flows, providing a more accurate estimate for x_0.

[1]The structure-independent information of the motion field is provided by (either of) the extrema of the projected flow. Since our goal is to extract this feature we do not lose relevant information in the collapsing step.

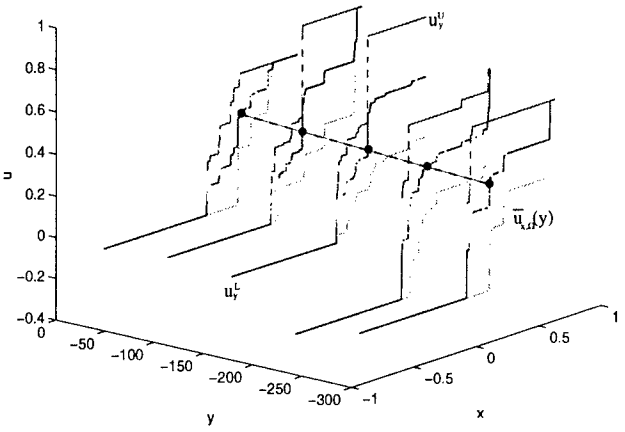


Figure 5: The collapsing of bands of $u(x, y)$ into a small number of lines (five in this case) results in a significant reduction in the number of parallel restrictions in the image. Each band provides a lower and upper distribution u^L, u^U from which $\hat{u}_\Omega$ is computed. The one-dimensional optimization in x gives $\hat{x}_0$ when a combination of line fitting and minimization of negative depth is employed.

The iteration steps can be implemented using a recursive observer model, where the state variables of the observer are the rotational parameters (Figure 6).

In order to employ this approach the stability of the recursive filter has to be assured. Applying the divergence property to parallel restrictions of large-FOV projected flow fields with rotational components results in erroneous estimates of x_0. This effect can be modeled as noise affecting the accuracy of the rotation estimates. It can be shown that this noise is bounded, where the bound is a function of the FOV. Expressing the stability conditions for the linearized discrete system model we can estimate the tolerance of the system to noise disturbances. This provides us with an upper bound on the allowable FOV ($> 100 - 120$ degrees) for which the recursive observer is stable. Experiments suggest that this bound is rather pessimistic and 2-4 iterations can usually approximate the motion parameters with less than 10-20% error.

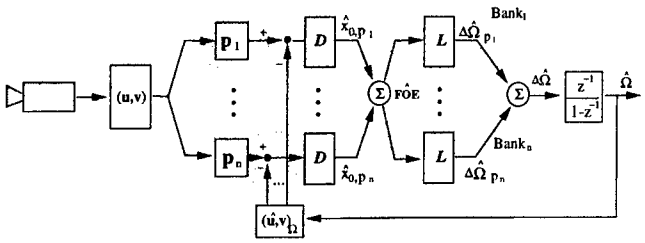


Figure 6: The recursive observer model implementing the divergence (D) and linearity (L) properties for the case of a large FOV. Each bank in the filter represents a specific projection direction. The vector components of the estimated motion parameters are combined in intermediate steps to provide the complete parameter vector.

4.4 Extension of the algorithm to an arbitrary FOE

Until now we have assumed that the projection of the FOE, x_0, is inside the image, or more precisely, inside the convex hull defined by the locations of the flow vectors. It is clear that neither the divergence nor the linearity properties can provide (unique) constraints on the visual motion if the FOE is outside this convex hull[2]. This is a theoretical limitation on the projective approach as compared with traditional methods; however, its practical shortcomings are minor.

The ambiguity of the divergence property can be illustrated as follows. If x_0 is not visible then the divergence property of parallel restriction $u_y(x)$ is satisfied by any point $(x, \hat{u}_{y,\Omega}(x))$ where $x < x_{\min}$ and $u_{y,\Omega}(x) < \min_x\{ u_y(x) \}$, or $x > x_{\max}$ and $u_{y,\Omega}(x) > \max_x\{ u_y(x) \}$, since the estimated depth at the location of each flow vector would be positive (Figure 4).

It follows that we cannot recover the accurate position of x_0 if it is not within the image. But we can still recover a *qualitative* estimate of x_0 which lies on the image boundary closest to x_0. In order to be able to recover this qualitative estimate we need to assume that the scene is sufficiently cluttered and has (approximately) constant variation ΔZ_0 for each x along the given y. The resulting variation in the parallel restriction of the translational projected flow is then $\Delta u_{y,T}(x) = \frac{|x-x_0|}{\Delta Z_0}$. For simplicity and without loss of generality we suppose that $x_0 < x_{\min}$ (i.e., x_0 lies beyond the left image border). It follows that for any $x \in (x_{\min}, x_{\max}]$ we have

$$\Delta u_{y,T}(x_{\min}) < \Delta u_{y,T}(x).$$

[2]This problem is relevant only if the translational component of the flow field is measurable; otherwise (e.g. $Z_0 = \infty$ or pure rotation) it is not possible to recover this component.

From the definition of the distribution functions (6,7) it follows that

$$u_y^U(x_{\min}) - u_y^L(x_{\min}) < u_y^U(x) - u_y^L(x).$$

Since our cost function (9) favors those x values for which the lower and upper distributions are closer to each other we get $\hat{x}_0 = x_{\min}$.

We will demonstrate in Section 5 that there exist navigation and structure-from-motion applications in which it is not necessary to recover an accurate value of x_0, but only to find a qualitative estimate of x_0 when it is not within the image.

5 Applications

In this section we demonstrate the application of our approach to a special qualitative structure-from-motion problem, the detection of independently moving objects.

Human and animal vision systems have the ability to detect moving objects even if the observer is moving. A human uses a combination of visual cues to solve this problem, such as geometry; shape; illumination; color; ordinal depth from stereo, structure and occlusion; knowledge about the scene; etc. As in many other studies [1; 7; 8; 10; 13], we use only the geometry cue to solve this problem, and assume no domain knowledge (which, of course, would be useful in practical applications). In our approach, the detection of moving objects is based on the rigidity analysis of visual motion fields using the constraint that depth is positive. In contrast to other work, we assume general camera motion and wide FOV, and we use the normal flow field.

5.1 Theory of detectability

Let $\boldsymbol{v}$ be a flow field and FOE the focus of expansion induced by the motion of the camera. We also define FOE_{mo}, which is induced by the relative motion of a moving object, ignoring the rigid scene. If FOE and FOE_{mo} are located at the same image point, then we cannot detect any inconsistency in $\boldsymbol{v}$ based on geometry, since the directions of the flow field are the same for both the rigid scene and the moving object[3]. If, however, FOE and FOE_{mo} are distinct, then there exist regions in the image plane where the flow vectors originating from the rigid scene and the ones from the moving object point in different directions (Figure 7a). In this case it follows that there always exists a projection direction, $\boldsymbol{p}$, for which the

[3]In general, when the directions are the same but only the magnitudes of the translational components vary, inconsistency cannot be detected unless we have partial depth information about the scene. For example, if we were able to detect *ordinal depth* [4], e.g. from occlusion, it would significantly improve our detection capabilities.

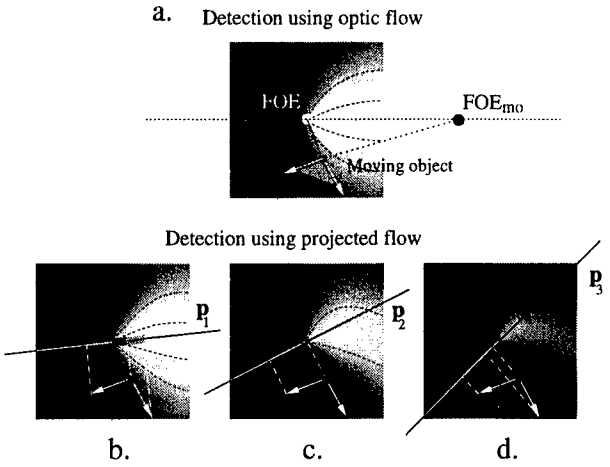


Figure 7: (a) Detectability of a moving object using optic flow. (b-d) Projections of optic flow in different directions. The brighter regions indicate easier detectability. In (c) and (d) the object is not detectable since the projected flow generated by the moving object lies in the same direction as that of the egomotion.

object is detectable from sign differences of the projected flow field. Based on this criterion, we can define ***p-detectability*** λ_p for a given image location $\boldsymbol{q}$ as

$$\lambda_p(\boldsymbol{q}) \stackrel{def}{=} (\boldsymbol{p}\boldsymbol{r}_{0,q})(\boldsymbol{p}\boldsymbol{r}_{mo,q}),$$

where $\boldsymbol{r}_{0,q}$ is the unit vector in the direction of $\boldsymbol{q}$ emanating from FOE and $\boldsymbol{r}_{mo,q}$ is the unit vector in the direction of $\boldsymbol{q}$ emanating from FOE_{mo}. It follows that if $\lambda_p(\boldsymbol{q}) < 0$, the moving object located at $\boldsymbol{q}$ can be detected from the single projection $\boldsymbol{p}$; otherwise, it is not detectable. The more negative $\lambda_p(\boldsymbol{q})$ is, the more significant the sign difference is, hence the easier is the detection of $\boldsymbol{q}$ as an independently moving point. We can also easily determine the optimal detection direction to maximize the size of the region in the image in which the moving object is detectable. This can be achieved when $\boldsymbol{p} \parallel \overline{FOE, FOE_{mo}}$ (Figure 7). These results suggest that it is not necessary to recover the complete motion of the camera to detect moving objects. If *a priori* knowledge is available about the possible motion of the moving target (e.g. detection of ground moving targets from a ground vehicle) then we can choose a single projection direction (in this case horizontal) from which the target is detectable.

5.2 Detection algorithm

Detecting moving objects requires identifying regions of actual independent motion and excluding regions of "outliers" [1]. Our detection algorithm is based on the estimation of the (projected) motion parameters of the camera; the motion estimation process is designed to tolerate inconsistencies in some of the bands of the projected flow field. Violation of the linearity property can be tolerated by using *robust* line-fitting based on a repeated-median algorithm [11] which assumes that moving objects corrupt the given projected flow in less than half of the total number of bands. It follows that the rather complex problem of detecting inconsistencies with rigidity in the visual scene is reduced to that of one-dimensional robust line-fitting and outlier detection. After this, the constraint of positive depth is examined by back-projecting the estimated projected flow. The regions where this constraint is vi-

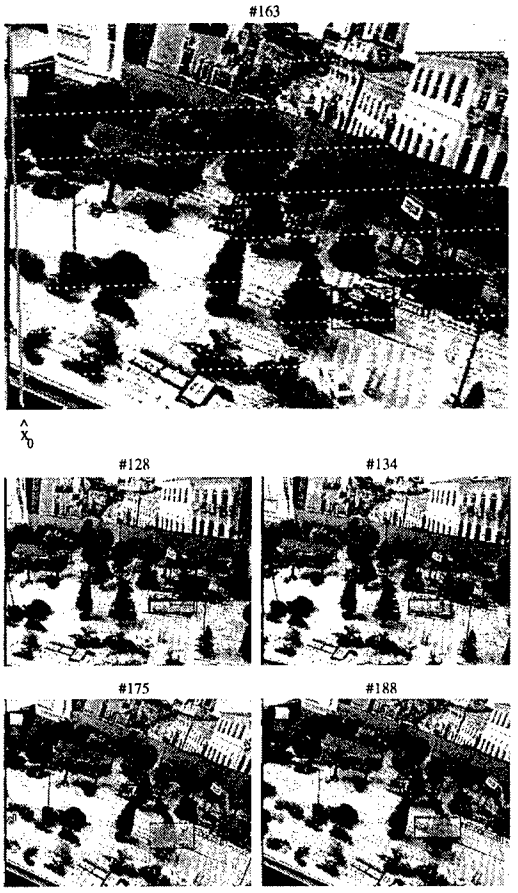


Figure 8: Detection of a toy car using normal flow and a single projection over a long image sequence. The hand-held, sideways-looking camera (~55-degree FOV) has general rotational and translational motion, and the FOE lies beyond the left image border. In the top image the dotted lines delineate the bands. The detected moving points were verified by temporal integration; the boxes in the bottom images represent groupings that are assumed to belong to the same rigid body. (For an MPEG demonstration see also http://www.cfar.umd.edu/~fejes/research_sum.html#CV)

olated are considered to be non-rigid, corresponding to independently moving objects.

There are several sources of error in the parameter estimation process, which practically require that the algorithm use certain thresholds. The choice of these thresholds is very critical, since a poor choice can cause either a high false alarm rate or a high ratio of undetected objects.

The outlier detection in the robust line fitting process ideally finds those bands corrupted by the presence of an independently moving object. We must then find the regions (pixels) within the corrupted bands that cause the inconsistencies, ideally with few false alarms. We employ an *adaptive threshold* which depends on the amount of error in the current motion estimate. If the estimate is erroneous, then the threshold has to be set high to avoid false alarms, whereas if it is accurate we can set the threshold low to achieve more sensitive detection.

The actual values of the motion parameters are, of course, not available; however, we can estimate the error by examining the degree to which the divergence and linearity properties are satisfied. If these properties are well satisfied then there is probably little error in the motion estimate. On the other hand, if these properties are violated (as is typically the case) we use as a measure of rigidity the largest error found in those bands which are not outliers. Using this value as the threshold results in a good trade-off between false alarms and false dismissals.

5.3 Integration of instantaneous measurements

Detection reliability also has to be considered at other levels of the model. Vision algorithms based on frame-by-frame analysis lack robustness unless they make use of temporal and spatial integration of the spatio-temporal visual information.

Temporal integration of our visual measurements is performed at two levels. The first level is realized by the recursive observer model which provides the estimation of the motion parameters. The second, higher level is realized using what we call the *visual scene memory*. This memory is the representation of motion-independent information, i.e., some kind of depth estimate, extracted from and continuously updated from motion fields of long image sequences. The cumulative maintenance of scene structure is achieved by *tracking* regions based on the estimated motion parameters. In our specific case of moving object detection, this memory is, at each time instant, binary and stores the sign of the estimated depth Z_0 at each scene point, which is negative only if the point is on an independently moving object. A point in the image is defined to be a *verified* target only if this status is sufficiently supported along the time axis. This model helps us (1) ignore false alarms which have short durations and (2) maintain detection in regions where short-term drop-outs occur.

Spatial integration of individual points is achieved by *grouping*. In our model, if two detected points are "close" to each other and have similar translational projected flow values, they are assumed to belong to the same rigid target. The smallest bounding box surrounding an equivalence class of this grouping relation is used to identify and represent a moving object (Figure 8).

6 Conclusions

In this paper a methodology is proposed which analyzes projections of visual motion fields. Taking advantage of the reduced dimensionality of projected subspaces, we identify two structure-invariant properties of projections of flow fields: the divergence and linearity properties. These properties serve as a basis for decoupling structure from motion and estimating projections of motion parameters, or if multiple projections are available, the complete set of motion parameters. The method does not require point correspondence; it is directly applicable to normal flow, and can handle wide-FOV flow fields. Our approach is implemented using a recursive filter which incorporates robust techniques. It is extended by using structure-related temporal and spatial integration, implemented in a model of visual scene memory. We have applied it to the problem of detecting independently moving objects from a moving camera.

Acknowledgments:

The authors would like to thank K. Daniilidis, Z. Duric and A. Rosenfeld for valuable comments and discussions.

References

[1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:384–401, 1985.

[2] K. Daniilidis and I. Thomas. Decoupling the 3D motion space by fixation. In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 685–696, Cambridge, UK, 1996.

[3] C. Fermüller and J. Aloimonos. Direct perception of three-dimensional motion from patterns of visual motion. *Science*, 270:1973–1976, 1995.

[4] C. Fermüller and J. Aloimonos. Qualitative egomotion. *International Journal of Computer Vision*, 15:7–29, 1995.

[5] C. Fermüller and J. Aloimonos. Algorithm-independent stability of structure from motion. Technical Report CAR-TR-840, University of Maryland at College Park, 1997.

[6] B.K.P. Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.

[7] M. Irani and P. Anandan. A unified approach to moving object detection in 2D and 3D scenes. In *Proceedings of the DARPA Image Understanding Workshop*, pages 707–718, Palm Springs, CA, 1996.

[8] C.H. Morimoto, D. DeMenthon, L.S. Davis, R. Chellappa, and R. Nelson. Detection of independently moving objects in passive video. In I. Masaki, editor, *Proceedings of the Intelligent Vehicles Workshop*, pages 270–275, Detroit, MI, 1995.

[9] R. Nelson and J. Aloimonos. Finding motion parameters from spherical motion fields (or the advantages of having eyes in the back of your head). *Biological Cybernetics*, 58:261–273, 1988.

[10] R. Sharma and J. Aloimonos. Early detection of independent motion from active control of normal image flow patterns. *IEEE Transactions on Systems, Man, and Cybernetics*, 26:42–52, 1996.

[11] A.F. Siegel. Robust regression using repeated medians. *Biometrika*, 69:242–244, 1982.

[12] C. Silva and J. Santos-Victor. Direct egomotion estimation. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 702–706, Vienna, Austria, 1996.

[13] W.B. Thompson and T.C. Pong. Detecting moving objects. *International Journal of Computer Vision*, 4:39–57, 1990.

A Kalman Filter That Learns Robust Models Of Dynamic Phenomena

Rajesh P. N. Rao*
Department of Computer Science
University of Rochester
Rochester, NY 14627-0226
E-MAIL: rao@cs.rochester.edu
HOMEPAGE: http://www.cs.rochester.edu/u/rao/

Abstract

We derive a Kalman filter that can autonomously learn an internal dynamic model of its input environment directly from the spatiotemporal input stream. The filter uses its learned internal model to maintain robust optimal estimates of the input environment's hidden state by allowing the measurement covariance matrix to be a non-linear function of the prediction errors. This endows the filter with the ability to reject outliers in the input stream. We present experimental results demonstrating the utility of such filters in appearance-based segmentation and recognition of objects and image sequences in the presence of varying degrees of occlusion and clutter.

1 INTRODUCTION

Three and a half decades after its discovery, the Kalman filter [Kalman, 1960] remains one of the most versatile algorithms in parameter estimation theory, having found applications in areas as diverse as economics [Athans, 1974], engineering [Cipra, 1993], and neuroscience [Rao and Ballard, 1996b]. One fundamental obstacle to the direct application of the Kalman filter to arbitrary state estimation problems has been the need to specify accurate dynamic models of the observed physical system. Furthermore, since it is derived from a least squares optimization criterion, the standard Kalman filter is highly susceptible to gross outliers in the input data stream.

*This work was supported by NIH/PHS research grant no. 1 P41 RRO9283.

In this paper, we describe how a Kalman filter can (a) autonomously learn an internal model of an observed dynamic system, and (b) reject outliers in the input stream, thereby allowing robust optimal estimation of the observed system's hidden state. The utility of these robust adaptive filters is illustrated using a relatively difficult problem in vision, namely, appearance-based segmentation and recognition of objects and image sequences, in the presence of varying degrees of occlusion and clutter.

2 THE KALMAN FILTER

The starting point for the derivation of the Kalman filter [Bryson and Ho, 1975] is the assumption that at time instant t, the dynamic process of interest is characterized by a $k \times 1$ internal state vector $\mathbf{r}(t)$ that cannot be measured directly, but generates measurable outputs $\mathbf{I}(t)$ in the following manner:

$$\mathbf{I}(t) = U\mathbf{r}(t) + \mathbf{n}(t) \qquad (1)$$

In the above, U is an $n \times k$ "measurement" matrix and the $n \times 1$ vector $\mathbf{n}$ is a stochastic white noise process with mean $E(\mathbf{n}) = 0$ and covariance $\Sigma = E[\mathbf{n}\mathbf{n}^T]$.

Given the state $\mathbf{r}(t-1)$ at time instant $t-1$, the next state $\mathbf{r}(t)$ is assumed to be given by:

$$\mathbf{r}(t) = V\mathbf{r}(t-1) + \mathbf{m}(t-1) \qquad (2)$$

where V is the *state transition (or prediction) matrix* and $\mathbf{m}$ is white Gaussian noise with mean $\overline{\mathbf{m}} = E[\mathbf{m}]$ and covariance $\Pi = E[(\mathbf{m} - \overline{\mathbf{m}})(\mathbf{m} - \overline{\mathbf{m}})^T]$.

Given the above model of the observed dynamic system, the goal is to optimally estimate the sys-

tem's hidden state $\mathbf{r}(t)$ using only the measurable inputs $\mathbf{I}(t)$. Suppose that we have already computed a prediction $\bar{\mathbf{r}}$ of the current state $\mathbf{r}$ based on prior data. In particular, let $\bar{\mathbf{r}}(t)$ be the mean of the current state vector *before* measurement of the input data $\mathbf{I}$ at the current time instant t. The corresponding covariance matrix is given by $E[(\mathbf{r}-\bar{\mathbf{r}})(\mathbf{r}-\bar{\mathbf{r}})^T] = M$. An optimization function whose minimization yields an estimate for $\mathbf{r}$ is the weighted least-squares criterion [Bryson and Ho, 1975]:

$$J = (\mathbf{I}-U\mathbf{r})^T\Sigma^{-1}(\mathbf{I}-U\mathbf{r}) + (\mathbf{r}-\bar{\mathbf{r}})^T M^{-1}(\mathbf{r}-\bar{\mathbf{r}}) \quad (3)$$

It is easy to show (see, for example, [Bryson and Ho, 1975]) that J is simply the sum of the negative log-likelihood of generating the data $\mathbf{I}$ given the state $\mathbf{r}$, and the negative log of the prior probability of the state $\mathbf{r}$. Thus, minimizing J is equivalent to maximizing the posterior probability $p(\mathbf{r}|\mathbf{I})$ of the state $\mathbf{r}$ given the input data (under the assumption that $p(\mathbf{r},\mathbf{n}) = p(\mathbf{r})p(\mathbf{n})$).

The optimization function J can be minimized by setting $\frac{\partial J}{\partial \mathbf{r}} = 0$ and solving for the minimum value $\hat{\mathbf{r}}$ of the state $\mathbf{r}$ (note that $\hat{\mathbf{r}}$ equals the mean of $\mathbf{r}$ after measurement of $\mathbf{I}$). The resultant *Kalman filter* update equation is given by:

$$\hat{\mathbf{r}}(t) = \bar{\mathbf{r}}(t) + N(t)U^T\Sigma(t)^{-1}(\mathbf{I}-U\bar{\mathbf{r}}(t)) \quad (4)$$

$$\bar{\mathbf{r}}(t) = V\hat{\mathbf{r}}(t-1) + \bar{\mathbf{m}}(t-1) \quad (5)$$

where $N(t) = (U^T\Sigma(t)^{-1}U + M(t)^{-1})^{-1}$ is a "normalization" matrix that maintains the covariance of the state $\mathbf{r}$ after measurement of $\mathbf{I}$. The matrix M, which is the covariance before measurement of $\mathbf{I}$, is updated as $M(t) = VN(t-1)V^T + \Pi(t-1)$.

3 A ROBUST FORM OF THE KALMAN FILTER

The standard derivation of the Kalman filter minimizes Equation 3 but unfortunately does not specify how the covariance Σ is to be obtained. A common choice is to use a constant matrix or even a constant scalar. Making Σ constant however reduces the Kalman filter estimates to standard least-squares estimates. It is well-known that least-squares estimation is highly susceptible to outliers or gross errors i.e. data points that lie far away from the bulk of the observed data [Huber, 1981]. For example, in the case where $\mathbf{I}$ represents an input image, occlusions and other forms of noise will cause many pixels in $\mathbf{I}$ to deviate significantly from corresponding pixels in the predicted image $U\mathbf{r}$. These components of $\mathbf{I}$ need to be treated as outliers and discounted for in the minimization process in order to get an accurate estimate of the state $\mathbf{r}$.

The problem of outliers can be tackled using *robust estimation procedures* [Huber, 1981]. One commonly used procedure is *M-estimation* (Maximum likelihood type estimation), which involves minimizing a function of the form:

$$J' = \sum_{i=1}^{n} \rho\left(\mathbf{I}^i - U^i\mathbf{r}\right) \quad (6)$$

where ρ is normally taken to be a less rapidly increasing function than the square. This ensures that large residual errors (which correspond to outliers) do not influence the optimization of J', thereby "rejecting" the outliers. Note that when ρ equals the square function, we obtain the standard least squares criterion. More interestingly, we obtain the following weighted least squares criterion also as a special case:

$$J' = (\mathbf{I}-U\mathbf{r})^T S(\mathbf{I}-U\mathbf{r}) \quad (7)$$

where S is a diagonal matrix whose diagonal entries $S^{i,i}$ determine the weight accorded to the corresponding data residual $(\mathbf{I}^i - U^i\mathbf{r})$. A simple but attractive choice for these weights is the non-linear function given by:

$$S^{i,i} = \min\left\{1, c/(\mathbf{I}^i - U^i\mathbf{r})^2\right\} \quad (8)$$

where c is a threshold parameter that can be modulated according to the application at hand. To understand the behavior of this function, note that S effectively clips the ith summand in J' to a constant value c whenever the ith squared residual $(\mathbf{I}^i - U^i\mathbf{r})^2$ exceeds the threshold c; otherwise, the summand is set equal to the squared residual.

By substituting $\Sigma^{-1} = S$ in the optimization function J (Equation 3), we can re-derive the Kalman filter update equations. The resulting *robust Kalman filter* for updating the state estimate is given by:

$$\hat{\mathbf{r}}(t) = \bar{\mathbf{r}}(t) + N(t)U^T G(t)(\mathbf{I}-U\bar{\mathbf{r}}(t)) \quad (9)$$

$$\bar{\mathbf{r}}(t) = V\hat{\mathbf{r}}(t-1)) + \bar{\mathbf{m}}(t-1) \quad (10)$$

where $N(t) = (U^T G(t)U + M(t)^{-1})^{-1}$, $M(t) = VN(t-1)V^T + \Pi(t-1)$, and $G(t)$ is an $n \times n$ diagonal matrix whose diagonal entries at time instant t are given by:

$$G^{i,i} = \begin{cases} 0 & \text{if } (\mathbf{I}^i(t) - U^i\bar{\mathbf{r}}(t))^2 > c(t) \\ 1 & \text{otherwise} \end{cases}$$

G can be regarded as the sensory residual gain or "gating" matrix, which determines the (binary) gain on the various components of the incoming sensory residual error. By effectively filtering out any high residuals, G allows the Kalman filter to ignore the corresponding outliers in the input $\mathbf{I}$, thereby enabling it to robustly estimate the state $\mathbf{r}$.

4 LEARNING A DYNAMIC MODEL

The measurement (or generative) matrix U and the state transition (or prediction) matrix V used by the Kalman filter together encode an internal model of the observed dynamic process. Most traditional applications of the Kalman filter employ hard-wired dynamic models inferred from a priori knowledge of the task at hand [Ayache and Faugeras, 1986, Blake and Yuille, 1992, Broida and Chellappa, 1986, Dickmanns and Mysliwetz, 1992, Hallam, 1983, Matthies *et al.*, 1989, Pentland, 1992]. These applications depend crucially on the ability to formulate accurate physical models of the object properties being estimated. In complex dynamic environments, the formulation of such hand-coded models becomes increasingly difficult.

An alternate approach is to *learn* an internal model of input dynamics directly from observed data, as suggested in [Rao and Ballard, 1996a]. Let $\mathbf{u}$ and $\mathbf{v}$ denote the vectorized forms of the matrices U and V respectively. For example, the $n \times k$ generative matrix U can be collapsed into an $nk \times 1$ vector $\mathbf{u} = [U^1 U^2 \ldots U^n]^T$ where U^i denotes the ith row of U. Note that $(\mathbf{I} - U\mathbf{r}) = (\mathbf{I} - R\mathbf{u})$ where R is the $n \times nk$ matrix given by:

$$R = \begin{bmatrix} \mathbf{r}^T & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{r}^T & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{r}^T \end{bmatrix} \tag{11}$$

By minimizing an optimization function similar to J (see [Rao and Ballard, 1996a] for details), one can derive a Kalman filter-based "learning rule" for the generative matrix U at time t:

$$\hat{\mathbf{u}} = \bar{\mathbf{u}} + N_u R^T G(\mathbf{I} - R\bar{\mathbf{u}}) \tag{12}$$

where $\bar{\mathbf{u}}(t) = \hat{\mathbf{u}}(t-1)$ and $N_u(t) = (N_u(t-1)^{-1} + R^T G(t) R)^{-1}$.

As in the case of U, an estimate of the prediction matrix V can be obtained via the following learning rule for $\mathbf{v}$ at time t [Rao and Ballard, 1996a]:

$$\hat{\mathbf{v}} = \bar{\mathbf{v}} + N_v \hat{R}(t)^T M^{-1}(\mathbf{r}(t+1) - \bar{\mathbf{r}}(t+1)) \tag{13}$$

where $\bar{\mathbf{v}}(t) = \hat{\mathbf{v}}(t-1)$, $N_v(t) = (N_v(t-1)^{-1} + \hat{R}(t)^T M(t)^{-1} \hat{R}(t))^{-1}$, and $\hat{R}(t)$ is the $k \times k^2$ matrix:

$$\hat{R}(t) = \begin{bmatrix} \hat{\mathbf{r}}(t)^T & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{r}}(t)^T & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & \hat{\mathbf{r}}(t)^T \end{bmatrix} \tag{14}$$

Note that in this case, the filter corrects its estimate of V using the prediction residual error $(\mathbf{r}(t+1) - \bar{\mathbf{r}}(t+1))$, which denotes the difference between the actual state and the predicted state.

An interesting question is the issue of convergence of the overall filtering/learning scheme involving $\mathbf{r}$, U, and V. Fortunately, one can appeal to the well-known Expectation-Maximization (EM) algorithm [Dempster *et al.*, 1977] and allow the overall scheme to converge by choosing appropriate values for the state $\mathbf{r}$ in the above learning rules for $\mathbf{u}$ and $\mathbf{v}$ (note that in the above rules, we did not specify values for $\mathbf{r}(t)$ (comprising $R(t)$) in Equation 12 and $\mathbf{r}(t+1)$ in Equation 13). The EM algorithm suggests that in the case of static input stimuli ($\bar{\mathbf{r}}(t) = \hat{\mathbf{r}}(t-1)$), one may use $\mathbf{r}(t) = \hat{\mathbf{r}}$ when updating the estimate for $\mathbf{u}$, where $\hat{\mathbf{r}}$ is the converged optimal state estimate for the given static input. In the case of dynamic (time-varying) stimuli, the EM algorithm prescribes the use of $\mathbf{r}(t) = \hat{\mathbf{r}}(t|N)$, which is the optimal temporally *smoothed* state estimate [Bryson and Ho, 1975] for time t ($\leq N$), given input data for each of the time instants $1, \ldots, N$. Unfortunately, the smoothed state estimate requires knowledge of future inputs and is computationally quite expensive. For the experimental results, we approximated the smoothed estimates by their online counterparts $\hat{\mathbf{r}}(t)$ when updating the matrices U and V during training.

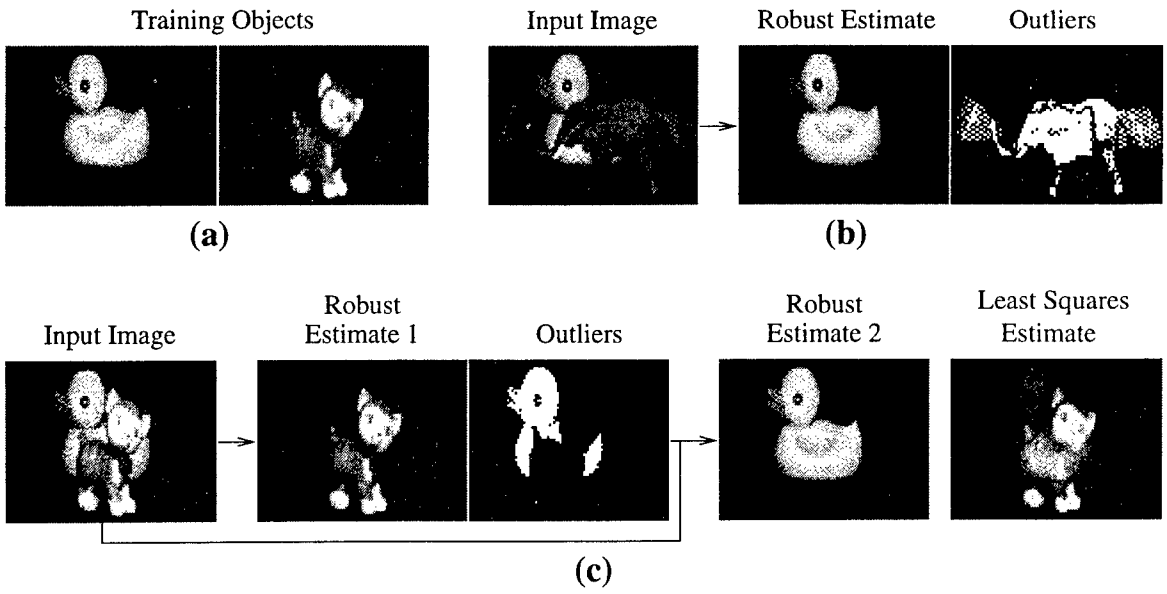


Figure 1: **Static Appearance-Based Recognition.** (a) Images used to train the robust adaptive filter. (b) Occlusions, background clutter, and other forms of noise are all treated as outliers (white regions in the third image, depicting the diagonal of the gating matrix G). This allows the filter to simultaneously segment and recognize the training object, as indicated by the accurate reconstruction (middle image) of the training image based on the final robust state estimate. (c) In the more interesting case of the training objects occluding each other, the filter converges to one of the objects (the "dominant" one in the image). Having recognized one object, the second object is recognized by taking the complement of the outliers (diagonal of G) and repeating the filtering process (third and fourth images). The fifth image is the image reconstruction obtained from the standard (least squares derived) Kalman filter estimate, showing an inability to resolve or recognize either of the two objects.

5 EXPERIMENTAL RESULTS

The robust adaptive filter derived above was applied to the problem of appearance-based recognition of objects and image sequences in the presence of occlusions and clutter. A prominent approach to appearance-based recognition is principal component analysis [Turk and Pentland, 1991, Murase and Nayar, 1995, Black and Jepson, 1996]. It is known [Rao and Ballard, 1996a] that the feedforward version of the Kalman filter-based learning rule for U is equivalent to Oja's principal subspace algorithm [Oja, 1989], which performs a form of principal component analysis. Thus, the Kalman filter based method described herein generalizes principal component (or *eigenspace* [Murase and Nayar, 1995]) based approaches by (a) allowing non-orthogonal basis vectors, (b) seeking more than pairwise correlations in the input data (when Equation 4 is augmented with a non-linear decay term [Olshausen and Field, 1996]), and (c) allowing learning and recognition of time-varying imagery.

In the first experiment, static grayscale images of size 65×105 depicting two $3D$ objects were used for training the filter (Figure 1 (a)). For learning static inputs, the prediction matrix V is unnecessary since we may use $\overline{\mathbf{r}}(t) = \hat{\mathbf{r}}(t-1)$ and $M(t) = N(t-1)$. After convergence of the filter for each input, the matrix U (of size 6825×5) was updated according to Equation 12. After training, the robust filter was tested on images containing the training objects with varying degrees of occlusion and clutter. The outlier threshold c was initialized to the sum of the mean plus k standard deviations of the current distribution of squared residual errors, decreasing k during each iteration. After convergence, the diagonal of the matrix G contains zeros in the image locations containing the outliers and ones in the remaining locations. As shown in Figure 1 (b), the filter was successful in segmenting and recognizing the training object in spite of considerable occlusion and background clutter. More interestingly, an *outlier mask* $\mathbf{m}$ can be defined by taking the complement of the diagonal of G (i.e. $\mathbf{m}^i = 1 - G^{i,i}$). By replacing the diagonal of G with $\mathbf{m}$ in Equation 9 and repeating the estimation process, one can obtain robust estimates of the image region(s) that were previously treated as outliers, thereby allow-

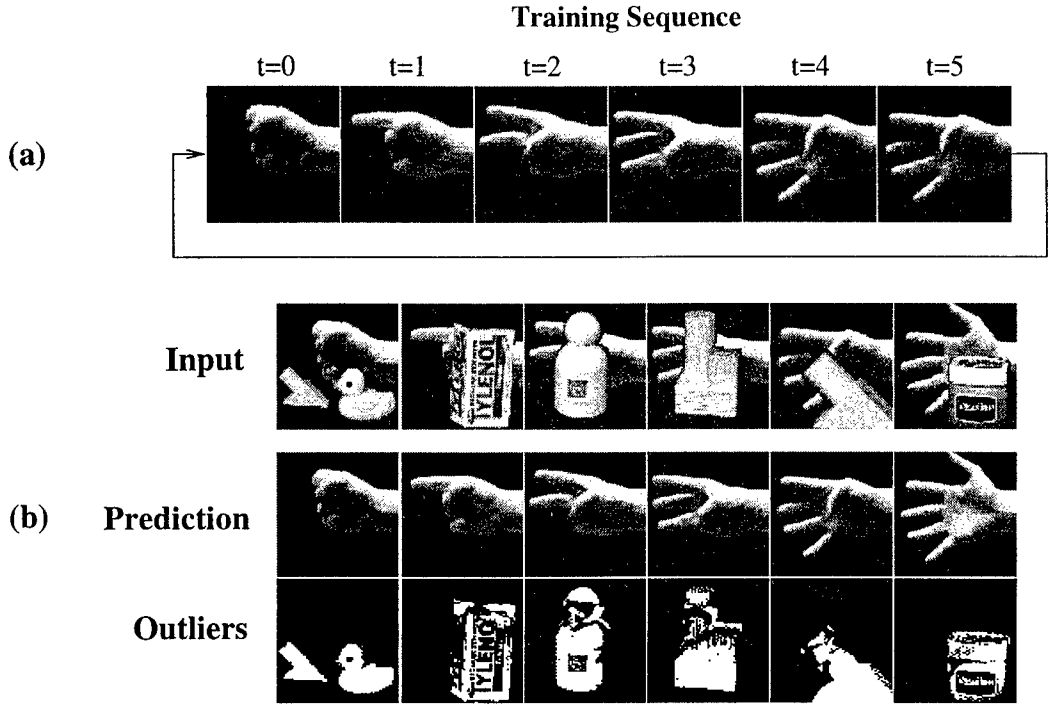


Figure 2: Dynamic Appearance-Based Recognition. (a) Cyclic image sequence of gestures used for training the adaptive filter. (b) Robust prediction and tracking of gestures in the presence of various forms of occlusions and clutter. Results shown are those obtained after five cycles of exposure to the occluded gesture images.

ing the filter to *recognize the occluder(s)* as shown in Figure 1 (c).

In a second experiment, a Kalman filter was trained on an image sequence depicting hand gestures (Figure 2 (a)). Each image was of size 75×75. The matrices U and V (of size 5625×15 and 15×15 respectively) were initialized to small random values, before training using Equations 12 and 13. During the recognition phase, the robustness parameter c was computed at each time instant as the sum of the mean plus 0.3 standard deviations of the current distribution of squared residual errors. As shown in Figure 2 (b), the filter was able to learn a sufficiently accurate dynamic model of the gesture sequence and use this model for robust recognition and tracking in the presence of various forms of occlusions and clutter.

6 DISCUSSION AND CONCLUSION

During the past decade, Kalman filters have been applied to a wide range of problems in computer vision [Blake and Yuille, 1992] and image processing [Chou *et al.*, 1994]. However, a majority of these approaches have used hard-wired dynamic models inferred from a priori knowledge of the task at hand. This paper suggests a relatively straightforward method for *learning* these dynamic models directly from input data, thereby avoiding the need for hand-coded physical models of the observed dynamic system. In addition, the robust Kalman filter proposed herein may serve as an alternative to more complex stochastic estimation schemes such as the CONDENSATION algorithm [Isard and Blake, 1996] for tackling the problem of occluders and background clutter in the input image stream. The robust Kalman filter presented here can also be readily extended to the hierarchical Kalman filter framework proposed in [Rao and Ballard, 1996a] and allows useful functional interpretations of neural circuitry in the visual cortex [Rao and Ballard, 1996b].

References

[Athans, 1974] M. Athans. The importance of Kalman filtering methods for economic systems. *Annals of Economic and Social Measurement*, 3:49–64, 1974.

[Ayache and Faugeras, 1986] N. Ayache and O.D.

Faugeras. HYPER: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(1):44–54, 1986.

[Black and Jepson, 1996] M.J. Black and A.D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *Proc. of ECCV*, pages 329–342, 1996.

[Blake and Yuille, 1992] A. Blake and A. Yuille, editors. *Active Vision*. Cambridge, MA: MIT Press, 1992.

[Broida and Chellappa, 1986] T.J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(1):90–99, 1986.

[Bryson and Ho, 1975] A.E. Bryson and Y.-C. Ho. *Applied Optimal Control*. New York: John Wiley and Sons, 1975.

[Chou *et al.*, 1994] K.C. Chou, A.S. Willsky, and A. Benveniste. Multiscale recursive estimation, data fusion, and regularization. *IEEE Transactions on Automatic Control*, 39(3):464–478, March 1994.

[Cipra, 1993] B. Cipra. Engineers look to Kalman filtering for guidance. *SIAM News*, 26(5), 1993.

[Dempster *et al.*, 1977] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society Series B*, 39:1–38, 1977.

[Dickmanns and Mysliwetz, 1992] E.D. Dickmanns and B.D. Mysliwetz. Recursive 3D road and relative ego-state recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):199–213, 1992.

[Hallam, 1983] J. Hallam. Resolving observer motion by object tracking. In *Proc. of 8th International Joint Conf. on Artificial Intelligence*, volume 2, pages 792–798, 1983.

[Huber, 1981] P.J. Huber. *Robust Statistics*. John Wiley and Sons, New York, 1981.

[Isard and Blake, 1996] M. Isard and A. Blake. Contour tracking by stochastic propogation of conditional density. In *Proc. of ECCV*, pages 343–356, 1996.

[Kalman, 1960] R.E. Kalman. A new approach to linear filtering and prediction theory. *Trans. ASME J. Basic Eng.*, 82:35–45, 1960.

[Matthies *et al.*, 1989] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.

[Murase and Nayar, 1995] H. Murase and S.K. Nayar. Visual learning and recognition of 3D objects from appearance. *IJCV*, 14:5–24, 1995.

[Oja, 1989] E. Oja. Neural networks, principal components, and subspaces. *International Journal of Neural Systems*, 1:61–68, 1989.

[Olshausen and Field, 1996] B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[Pentland, 1992] A.P. Pentland. Dynamic vision. In G.A. Carpenter and S. Grossberg, editors, *Neural Networks for Vision and Image Processing*, pages 133–159. Cambridge, MA: MIT Press, 1992.

[Rao and Ballard, 1996a] R.P.N. Rao and D.H. Ballard. Dynamic model of visual recognition predicts neural response properties in the visual cortex. *Neural Computation*, 9(4):805-847, 1997. Also, Technical Report 96.2, National Resource Laboratory for the Study of Brain and Behavior, Department of Computer Science, University of Rochester, 1996.

[Rao and Ballard, 1996b] R.P.N. Rao and D.H. Ballard. The visual cortex as a hierarchical predictor. Technical Report 96.4, National Resource Laboratory for the Study of Brain and Behavior, Department of Computer Science, University of Rochester, September 1996.

[Turk and Pentland, 1991] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

LEARNING TO FIXATE ON 3D TARGETS WITH UNCALIBRATED ACTIVE CAMERAS

Narayan Srinivasa and Narendra Ahuja
Beckman Institute and Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

Abstract

Given a target, fixation of an active camera pair requires that the pan and tilt angles must be set to bring the target to image centers. This paper defines a direct mapping from the changes in the direction of target motion in the image plane to changes in camera angles necessary to reduce the image plane disparity between image center and the target location. The mapping captures camera calibration and unmodelled effects such as deviations from the assumed imaging model. The mapping is formulated as a task in nonlinear function approximation, and, for computational efficiency, learnt from real data at multiple resolutions. In this work the learning is performed using a neural network. Experimental results are presented using an active vision system.

1 Introduction

For active vision systems to be useful for performing real tasks, it is critical that the camera control and processing be real-time. The increased availability of powerful and cheap computers and real-time image processing capabilities are making such systems feasible. This paper is aimed at the capability of fixation which is an integral part of active visual analysis [2, 3, 4, 5, 13]. To fixate on a three-dimensional (3D) point, the orientations (joint angles) of the cameras are changed such that the optical axes of the stereo camera move to intersect at the 3D point.

It is possible to compute an exact analytical expression for the camera joint angles required to fixate on a 3D point. This requires an accurate calibration of the various camera parameters [14]. The calibration process is usually tedious and time consuming. However, complete calibration is more than what is required to fixate. It suffices to know how to continuosly approach the state of fixation from current camera position using the images obtained during the fixation as feedback instead of directly transitioning to the state of fixation in one reconfiguration step. This paper uses a Direction-to-Joints (DTJ) mapping which models the relationship between incremental changes in the direction of image plane motion of the scene points and incremental changes in joint angles. DTJ exploits the property that for a given image of a 3D target, as camera joint angles are changed by a small amount, the direction in which the image of the target moves is independent of the current joint angles and the 3D target location.

We present a neural network based approach to learn the DTJ mapping at multiple resolutions of incremental camera motions. Initially, the coarse resolution DTJ mapping is used to rapidly bring the target roughly to the vicinity of the camera. Then, increasingly fine resolution DTJ mappings are used to monotonically reduce the residual error to accurately fixate on the 3D target. An interesting aspect of the approach is that a *single* target at a *fixed depth* from the active camera is sufficient to learn the DTJ mapping over the entire joint space for which the target is visible. Thus, it is easy to implement the learning in an autonomous mode on a real active vision system. The learning process presented is also self-organizing because all the training inputs and outputs are self-generated. Finally, as stated earlier, the learning approach does not require calibration.

2 Definitions

The most popular form of experimental setup used in active vision research has two motorized cameras mounted as a 'head' [1, 5, 7, 8]. The University of Illinois Active Vision System (UIAVS) [1] is one such system. Typically, each camera is mounted on a platform and controlled by a separate motor and enables the camera to move from side to side. The platform is itself motorized such that both the cameras have a common tilt angle. This allows the camera to assume arbitrary azimuth and elevation angles. In this paper, the UIAVS will be used to perform all the experiments.

A simplified kinematic model of the UIAVS is shown in Figure 1. The angles q_1 (and q_3) represent the angular positions of optical axis of the left (right) camera. These angles are called *pan angles* and are defined with respect to a "straight-ahead" direction (X-axis). The *tilt* is represented by the angle q_2 between the plane de-

fined by the optical axes and the XY plane. The direction conventions for each of these three angles are shown in Figure 1. These three angles are independently controlled. The point of intersection of the optical axes is called the *fixation point* (P in Figure 1). When a scene point is fixated, its image appears at the image centers for both cameras.

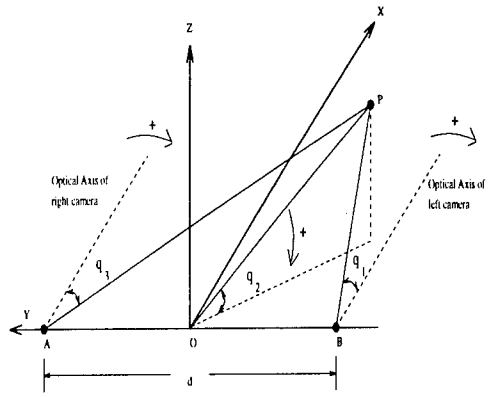


Figure 1: Simplified schematic of the UIAVS.

3 Existence of DTJ mapping

Let us assume that we have a calibrated active camera system with the following characteristics: the image is formed according to the pin-hole imaging model, the pan and tilt axes of each camera pass through its optic point, and the camera coordinate frames (centered at A and B in Figure 1) and the coordinate frame of the active vision system (centered at O) are offset by half of the baseline distance d. Let a 3D point P be imaged at (u, v) for the position (q_1, q_2) of the left camera. Then, for an incremental change $(\Delta q_1, \Delta q_2)$ in the camera orientation, the corresponding change in the image position $(\Delta u, \Delta v)$ can be derived using the image jacobian [9] as,

$$\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} \frac{\lambda^2+u^2}{\lambda} & \frac{-uv}{\lambda} \\ \frac{uv}{\lambda} & \frac{-\lambda^2-v^2}{\lambda} \end{bmatrix} \begin{bmatrix} \Delta q_1 \\ \Delta q_2 \end{bmatrix} \quad (1)$$

where λ is the focal length of the camera.

In order to establish the existence of the DTJ mapping, let us rewrite the left hand side of equation (1) in terms of the image plane direction components (n_x, n_y) where $n_x = \frac{\Delta u}{C}$ and $n_y = \frac{\Delta v}{C}$ and $C = (\Delta u^2 + \Delta v^2)^{\frac{1}{2}}$. The magnitude C depends upon the depth of the target from the active camera (closer the target, larger the magnitude) as well as the magnitude of $(\Delta q_1, \Delta q_2)$. The direction vector (n_x, n_y) depends on (u, v) and the ratio $\frac{\Delta q_1}{\Delta q_2}$. To illustrate this, a simple experiment was performed using an uncalibrated UIAVS. A set of random 3D targets (small dark patches on a white background) at various 3D locations were viewed one at a time by the UIAVS using a variety of camera joint angles. For each pair of 3D target location and camera orientation, the camera pan and tilt angles were changed over a wide range at fixed angular increments ($\Delta q_1 = \Delta q_3 = 0.001°$ and $\Delta q_2 = 0.003°$ where the resolution of the joint encoders of the pan and tilt units is 0.001°). The observed directions of target motion in image were recorded for small neighborhoods of (u, v) values of target location, regardless of 3D target location and camera joint angles. Results for seven such neighborhoods in the left camera image are shown in Figure 2. As can be seen in Figure 3, the image plane directions are similar for an entire neighborhood, and are different for the different neighborhoods (appearing as seven different curve segments for seven different clusters of (u, v) values). This serves as empirical evidence for the existence of the DTJ mapping.

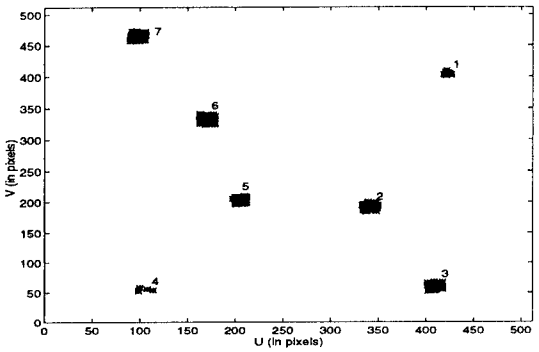


Figure 2: Seven clusters of image coordinates.

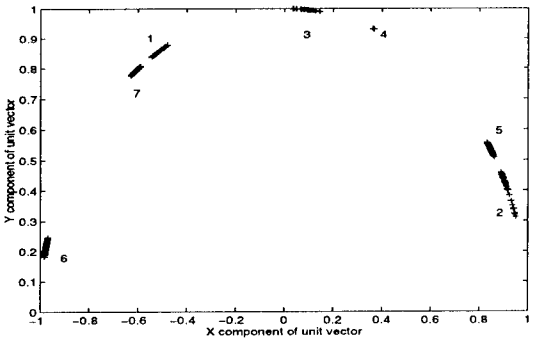


Figure 3: Seven clusters of the unit directional vectors.

4 Estimation of DTJ mapping

A straight forward approach to estimating DTJ for a given active vision system will be to estimate the 2X2 matrix in equation (1). This could be done by obtaining a least squares estimate of λ from observations of u and v and $(\Delta u, \Delta v)$ values for many Δq_1 and Δq_2. However, equation (1) is for the ideal case. In practice, the rotation axes of an active camera may be offset from the optic point. This will result in additional parameters in the Jacobian, which will also need to be estimated. Any deviations from the pin-hole model of image will lead to yet other unknowns. Therefore, we view this complicated mapping function from image plane directions to changes in camera joints as a non-linear function approximation task.

To obtain the data for learning, a single target at a fixed depth suffices, provided it is visible to the camera at all orientations. An additional benefit of using a single target is that the training examples for learning can be generated automatically making the implementation easier. The learning yields samples of mapping for selected (u, v, n_x, n_y) values for which $(\Delta q_1, \Delta q_2)$ was generated. Mapping for other intermediate values of (u, v, n_x, n_y) are obtained using the interpolation capabilities of the neural network.

5 Implementation of Multi-Resolution DTJ Mapping

The DTJ learning is posed as a function approximation problem. The inputs of the function are (u, v, n_x, n_y) while the corresponding outputs are $(\Delta q_1, \Delta q_2)$. The learning process is presented with several examples of these inputs and outputs and the DTJ mapping is approximated from these examples. We perform learning at a range of step sizes (resolutions) of $(\Delta q_1, \Delta q_2)$. This is because while using the mapping to determine the $(\Delta q_1, \Delta q_2)$ for a desired image plane motion direction, it may be desirable to perform the $(\Delta q_1, \Delta q_2)$ control in a coarse-to-fine manner for computational efficiency. The large camera motions (or *ballistic mode* of fixation) will bring the target image roughly near the image center, and, the fine resolution mapping will help bring the target into fixation more accurately.

In this paper, we use the PROBART neural network [11] for learning (Figure 4). This network is capable of incremental function approximation of non-linear mappings.The main advantage of this network over other commonly used networks such as the back-propogation [12] and Kohonen [10] is that it is capable of retaining the knowledge from previously presented inputs but at the same time accomodate knowledge from new inputs in an incremental fashion. The network consists of two clustering modules one of which is used to cluster the inputs to the function while the other is used to cluster the corresponding outputs of the function. These two modules are then mapped by weights F that measure the frequency of co-activation of the winning input and output clusters.

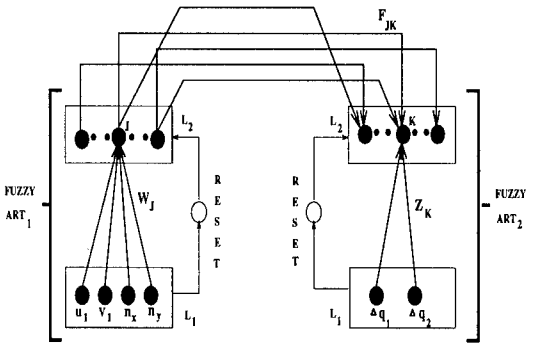


Figure 4: The PROBART neural network architecture.

To achieve multi-resolution performance, we use one PROBART network for each resolution. It consists of two Fuzzy ART networks [6] as shown in Figure 4. The Fuzzy ART network is capable of unsupervised and stable clustering of both binary and analog inputs in real-time. The Fuzzy ART_1 is used as the input processing module. At each resolution, the input layer L_1 module receives two types of inputs: (1) Image coordinates (u_1, v_1) of the non-fixated 3D point target on the left camera and the (2) Image direction (n_x, n_y). The L_1 layer of the Fuzzy ART_2 module receives the change in camera pan and tilt $(\Delta q_1, \Delta q_2)$ that caused the image of a 3D target at (u_1, v_1) to move in (n_x, n_y) direction. These inputs are then clustered in the L_2 layers of the two Fuzzy ART modules. For brevity, the clustering process for the Fuzzy ART will not be described here. The reader is referred to [6] for complete details on the clustering process.

During the clustering process, the weights between the two Fuzzy ART networks are updated by incrementing F_{JK} by 1 where J and K are the winning clusters in the two ART modules. It should be noted that the initial values of F are zero for all j and k and that the learning process for the right camera is the same as for the left camera. Once all the inputs and outputs are presented to the PROBART network, its predicting ability can be tested using inputs not seen during the training period. An input is first presented to the L_1 layer of the Fuzzy ART_1 network. A cluster J whose weight W_J that is most parallel to the input is selected as the winner. Then, the weights F are used to predict changes in the joint camera position as follows:

$$\Delta q_i = \frac{\sum_{k=1}^{R} F_{Jk} Z_{ik}}{\sum_{k=1}^{R} F_{Jk}} \quad (i = 1, 2) \qquad (2)$$

where Z_{ik} are the weights for the cluster k in the Fuzzy ART_2 module and R is the number of clusters in the L_2 layer of the Fuzzy ART_2 module.

5.1 Training

The regimen to learn the multi-resolution DTJ mapping using the PROBART network is outlined in Figure 5. The UIAVS is moved to various non-fixated camera positions for which a single target at a fixed depth is viewed. The location of the target is chosen such that the image of the target is visible for all values of the pan and tilt angles. These non-fixated camera positions are obtained using a random generator of pan and tilt angles within the allowable joint ranges. At each of these non-fixated camera positions, the pan and tilt of the camera are randomly moved in increments of multiple increments to generate the training inputs $V_k = (u_1, v_1)$ and $D = (n_x, n_y)$ and outputs $Q_k = (\Delta q_1, \Delta q_2)$ at each time step k as shown in Figure 5. The direction vector D is computed by using the image coordinates of the fixed target at two consecutive time steps.

and tilt angles. These non-fixated camera positions are obtained using a random generator of pan and tilt angles within the allowable joint ranges. At each of these non-fixated camera positions, the pan and tilt of the camera are randomly moved in increments of multiple increments to generate the training inputs $V_k = (u_1, v_1)$ and $D = (n_x, n_y)$ and outputs $Q_k = (\Delta q_1, \Delta q_2)$ at each time step k as shown in Figure 5. The direction vector D is computed by using the image coordinates of the fixed target at two consecutive time steps.

5.2 Performance Evaluation

Once the PROBART network has been trained on the DTJ mapping at multiple resolutions of joint angle increments, it is possible to use the learned mapping to fixate on any randomly selected 3D target. There are two modes in which the fixation can occur. The first mode is based on the availability of a *continuous visual feedback* as shown in Figure 6. The role of the incremental motion generator during training (Figure 5) is now replaced by the trained neural network. In this mode, a randomly selected 3D point target is first viewed by the

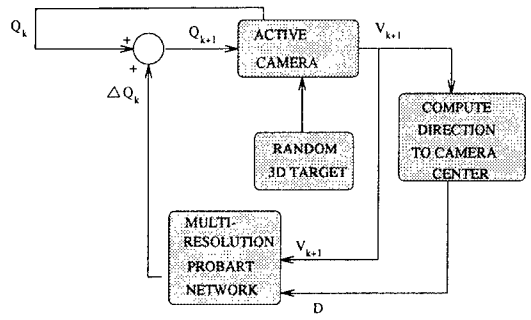


Figure 6: Continuous visual feedback approach.

active camera at some non-fixated camera position. In order to fixate on a 3D target, the coarse resolution network first generates large increments in the joint angles of the camera. The desired direction for each camera is obtained from the vector connecting the current image coordinate (V_k at time step k) of the 3D target and the image plane center. The coarse resolution network is used as long as the distance between the current image position and the image center is greater than the minimum distance that can be commanded by using the current resolution. When this condition is violated, the control is transferred to a network at the next resolution. In this manner, the incremental changes in joint angles at time k (ΔQ_k) are generated by an appropriately chosen PROBART network until the target is accurately fixated.

The second mode of the fixation process is based on the availability of only an *intermittent visual feedback* as shown in Figure 7. At the initial time step, the image coordinate of the target is known. Using this information, the direction D to the camera center is computed. By assuming a constant magnitude M (fixed different

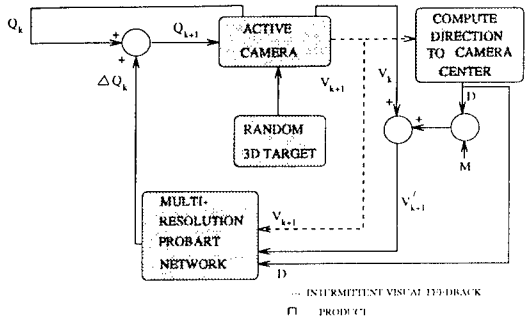


Figure 7: Intermittent visual feedback approach.

for different resolutions) of image motion, the new image location V'_{k+1} due to the change in joint angle ΔQ_k is predicted. The process continues until the predicted V'_{k+1} is within 0.5 pixels of the image center. When this occurs, a visual feedback is provided to verify if the camera has actually fixated on the target. If not, the actual image location is used to reiterate the fixation process until the target is fixated.

6 Experiments and Results

The PROBART neural networks at multiple resolutions were interfaced with the camera joint actuators and the image signals from the UIAVS. The active vision setup is mounted on a mobile robot to provide additional mobility. However, the mobile robot was not used during any of our experiments. The camera motion can be controlled by the tilt and pan units. The experimental setup for training the PROBART network is shown in Figure 8. A single dark patch was placed on a wall at a

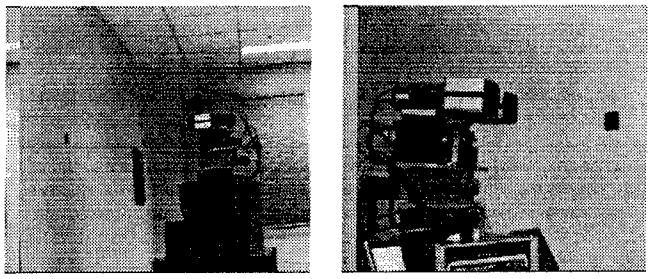


Figure 8: Experimental setup during training.

depth of about 4 meters from the active vision system. The image of the patch was extracted by thresholding and its centroid was used as the target.

In the experiments reported here, the DTJ mapping was learned at two resolutions. The angle changes at the coarse resolution for pan and tilt angles was randomly selected to be within 0.5° and 1° while those for the fine resolution were selected to be between 0.001° and 0.003°. The resolution of the joint encoders is 0.001°. The allowable joint angle range for both the pan and tilt angles was [−60°, 60°]. Since the image contains 512 X 512 pixels, each coordinate was normalized to the range

[0, 512]. The focal length λ was fixed at $30mm$. The regimen outlined in section 5.1 was adopted to train the PROBART neural networks.

Training data was collected at 60,000 camera orientations were collected. At each of these orientations, the active camera was incrementally moved both by coarse and fine camera motions. For each such motion, the (n_x, n_y) and $(\Delta u, \Delta v)$ were sampled. This training data was clustered by the coarse resolution network (of each camera) into 302 input and 16 output clusters. Similarly, the fine resolution networks created 1231 and 49 clusters for the inputs and outputs respectively.

Once the networks were trained on the inputs and outputs generated by observing a single target, the performance of the trained network was evaluated by placing 3D targets one at a time in a 4x4x6 cubic meter volume in front of the active camera. Each of these targets were fixated using the continuous and intermittent visual feedback modes. The image and joint trajectories during fixation in the continuous mode are shown in Figure 9 for the left camera and for a target at $(X = 3.5, Y = 2.0, Z = 2.0)$. This plot corresponds to the prediction of the multi-resolution PROBART network. The image is rapidly brought to within 6 pixels of the image center by the coarse resolution network. Then, the fine resolution network brings the target accurately into fixation. An accuracy of 0.05 pixels was obtained for all the targets. The average time taken to bring each of these targets was about 6 secs using the multi-resolution networks.

In order to compare the multi-resolution approach to just a single JTD mapping at fine resolution, the above plots were repeated without the coarse resolution JTD mapping. These plots are shown in Figure 10. It can be seen that the time taken to bring the same target into fixation is 20 times more than in the multi-resolution case. The accuracy of fixation is however the same for single and multi-resolution cases.

For the intermittent mode of visual feedback, the time taken to fixate on a target clearly depends on the accuracy of the mapping. The main parameter that affects this accuracy is the constant M (Figure 7). In the multi-resolution case, M can be large (for the coarse resolution) or small (for the fine resolution).

However, for large values of M, the system can have large overshoots and thus take prohibitively long periods of time to fixate. These overshoots can also be detrimental to the camera control equipment. To prevent this, we used the active camera only for small values of M. For comparison, the image and joint trajectories for the same 3D target are plotted for the intermittent mode as shown in Figure 11. The value of M was set to 0.05 pixels. This setting was reasonable because it prevented overshoots in the image trajectory for all the 3D targets. The inflection points in the image trajectory correspond to instances of visual feedback during the fixation process as shown in Figure 11(i). For all the targets in our experiments, the maximum number

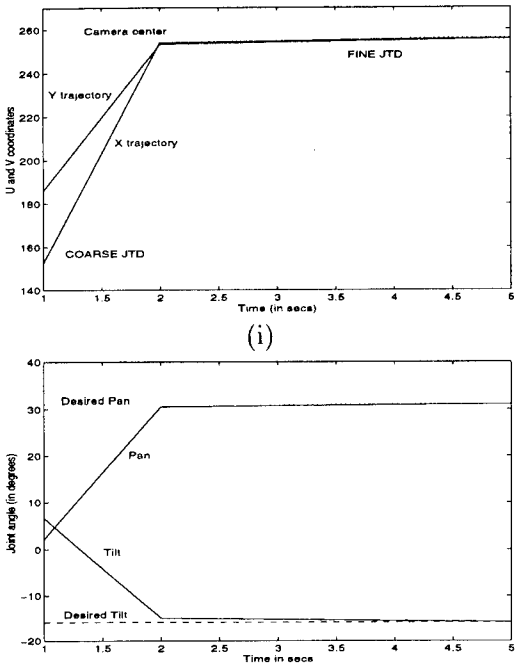


Figure 9: Performance of multi-resolution network during continuous visual feedback for the left camera (i) image trajectories and (ii) camera joint trajectories

of feedbacks was less than 10. It can be seen from Figures 11 that the intermittent mode takes longer compared to even the single resolution mode. However, there are no overshoots and the accuracy of fixation is not compromised.

7 Conclusions

There has been considerable interest in applying active vision to various image analysis tasks. This interest is primarily because active vision can enhance the capabilities of machine vision by dynamically changing the camera parameters. In this paper, we address the issue of learning to fixate on 3D point targets. This is achieved by exploiting a Direction-To-Joints or DTJ mapping that relates camera motion to image motion. The learning does not require the camera to be calibrated. Once the DTJ mapping is learned for multiple resolutions, it is possible to rapidly fixate on any other visible 3D target. This is achieved by using the learned DTJ mapping in a control loop with continuous or intermittent visual feedback. Experiments were performed on the UIAVS to verify the feasibility and accuracy of the proposed approach. The results obtained suggest that our approach is accurate and easy to implement on a real active vision system.

References

[1] L. Abbott and N. Ahuja. Surface reconstruction by

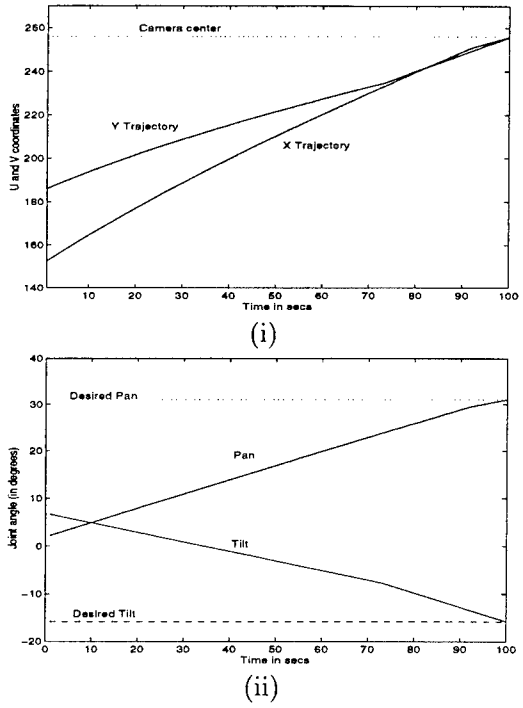


Figure 10: Performance with continuous visual feedback using only the fine resolution JTD mapping for the left camera (i) image trajectories and (ii) camera joint trajectories

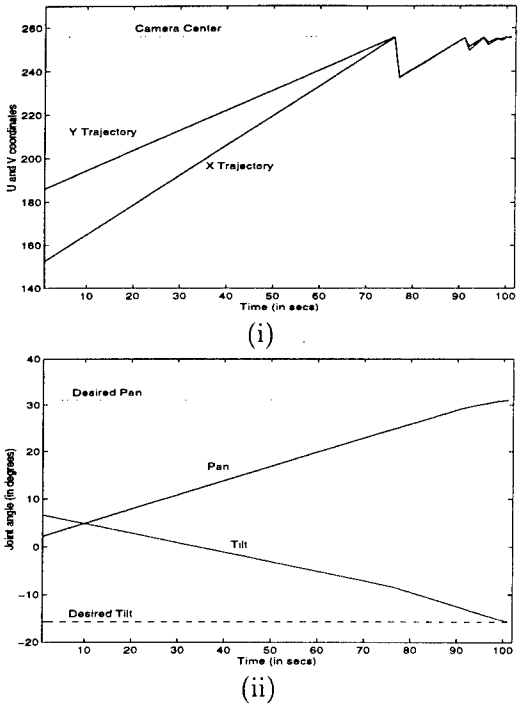


Figure 11: Performance with intermittent visual feedback for the left camera (i) image trajectories and (ii) camera joint trajectories

dynamic integration of focus, camera vergence and stereo. In *Proc. IEEE International Conference on Computer Vision*, pages 532–543, 1988.

[2] N. Ahuja and A. L. Abbott. Active stereo: Integrating disparity, vergence, focus, aperture, and calibration for surface estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1007–1029, 1993.

[3] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1:333–356, 1988.

[4] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 78:996–1005, 1988.

[5] D. Ballard and C. Brown. Principles of animate vision. In Y. Aloimonos, editor, *Active Perception*. Hillsdale, N.J. : Lawrence Erlbaum Associates, 1993.

[6] G. A. Carpenter, S. Grossberg, and D. B. Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.

[7] N. J. Ferrier. The harvard binocular head. Technical Report 91-8, Harvard Robotics Laboratory, 1991.

[8] F. Fuma, E. P. Krotkov, and J. Summers. The pennsylvania active camera system. Technical Report MS-CIS-86-15, GRASP Laboratory, University of Pennsylvania, 1986.

[9] S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, 1996.

[10] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[11] S. Marriott and R. F. Harrison. A modified fuzzy artmap architecture for the approximation of noisy mappings. *Neural Networks*, 8:619–641, 1995.

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing*, volume 1. MIT Press, 1986.

[13] M. J. Swain and M. Stricker. Promising directions in active vision. Technical Report CS 91-27, University of Chicago, 1991.

[14] G.-Q. Wei and S. D. Ma. Implicit and explicit camera calibration: Theory and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:469–480, 1994.

Temporal Multi-scale Models for Image Motion Estimation

Yaser Yacoob and Larry S. Davis
Center for Automation Research
University of Maryland, College Park, MD 20742-3275

Abstract

A model for computing image flow in image sequences containing a very wide magnitude range of instantaneous flows is proposed. This model integrates the spatio-temporal image derivatives from multiple temporal scales to provide both reliable and accurate instantaneous flow estimates. The integration employs robust regression and automatic scale weighting in a generalized brightness constancy framework. In addition to instantaneous flow estimation the model supports recovery of dense estimates of image acceleration. A demonstration of performance on image sequences of typical human actions, taken with a high-frame-rate camera, is given.

1 Introduction

Image motion estimation involves relating spatial and temporal changes in image intensity to estimates of image flow. Articulated and deformable motions such as those encountered in images of humans in motion give rise to image sequences having, simultaneously, a wide range of flow magnitudes ranging from very small sub-pixel motions whose recovery is inhibited by typical signal-to-noise constraints, to very large multiple-pixel motions whose recovery requires expensive correlation methods or multi-resolution approaches. Here, we focus on the problem of estimating dense image flow for image sequences in which the instantaneous flows range from 2-4 pixels/frame to $1/16-1/32$ pixel/frame. The difficulty is that we do not know a priori which parts of the image are moving with which speed. Our solution is a scale-space like solution [7] in which we estimate image flow over a wide range of temporal scales, and combine these estimates (using both spatial and temporal constraints) using a combination of robust estimation and parametric modeling as in [4].

The support of the Defense Advanced Research Projects Agency and the Office of Naval Research under Grant N00014-95-1-0521 is gratefully acknowledged.

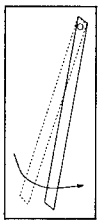


Figure 1: Pendulum movement illustrating varying velocities along its motion path

To motivate both the problem and our proposed solution, consider a pendulum arm moving in front of a camera. The image flow will vary depending on the distance of the measured point from the hanging point (see Figure 1). As we move towards the hanging point the instantaneous flow becomes very small and falls into the noise range of the imaging system. As a result, two-frame estimation and subsequent integration of these flow measurements over time will be highly noisy. In the context of human motion, the coincidence of lip motion with body and head motion, or the rotation of the calf around the knee, create similar scale variations in the flow field.

The majority of published algorithms for estimation of image flow are based on image pairs (for a recent survey see [1]). Several approaches, however, consider the incremental estimation of flow [3, 9]; then, temporal continuity of the flow applied over a few images (for example, assuming constant acceleration) can improve the accuracy of the flow estimates. These approaches are based on computations between consecutive images. Other approaches use velocity-tuned filters (i.e., frequency-based methods) [5, 6] to compute the flow, and can be extended to flow estimation from several frames. The use of scale-space theory to compute optical flow was recently proposed by Lindeberg [8]. The proposed algorithm focused on scale selection in the spatial dimension so that different-size image structures lead to different selections of scales for flow computation. The algorithm estimates flow from two images and

involves spatial multi-scales.

The approach presented in this paper simultaneously estimates

- small and large flows (spatially and temporally)
- dense flow and acceleration.

This paper is organized as follows. Section 2 illustrates, using an image sequence, the inadequacy of single-scale flow estimation. In Section 3 we describe the motion model employed for estimating image flow from multiple scales; this is followed by experimental results in Section 4. Section 5 provides the extension of the model to compute image acceleration. Finally, in Section 6 a discussion and a summary of our approach are provided.

2 A Motivating Example

We will use *scale*=1 to denote flow estimation between two consecutive images (i.e., the finest temporal resolution available),*scale* =2 to denote flow estimation between images that are two frames apart, etc. To illustrate the limitation of image flow estimates from any single scale we employ an image sequence of an arm moving in front of a camera. The sequence was taken with a high-frame-rate camera (500 frames per second) which allows us to capture the natural rapid motion of the arm. The arm (see Figure 2) is moving in a pendulum-like motion (with the hand rotating around the arm during the motion) in front of a lightly textured background*. Notice that there is a shadow created by the hand, leading to non-zero flow estimates of the shadow as well as the arm. The arm's intensity pattern consists of two parts: the arm itself is highly textured (allowing better flow estimation) while the hand is relatively uniform in brightness. Figure 2 shows eight images from the sequence (chosen two frames apart). The motion of the arm between two frames is very small, but it will become apparent when the flow estimates are shown.

Figure 3 shows the image flow magnitudes for six scales (falling on a geometric scale 1,2,4,8,16, and 32 frames apart). The finest scale provides detailed estimates of the flow magnitude at the hand but quite noisy estimates along the arm, while the coarsest scale results in accurate estimates along the arm but considerably blurred and inaccurate estimates on the hand.

*The intensity variation along the boundaries of the quadrants of the background is because the video camera has four separate A/D banks. As a result, flow estimation at the quadrant boundaries is inaccurate. This problem could be overcome by local gain compensation.

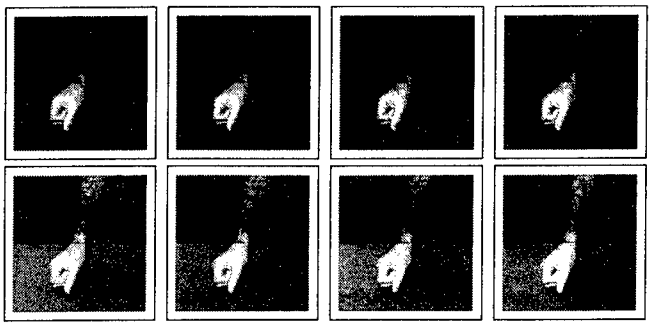


Figure 2: Eight images (two frames apart) from a long sequence of a moving arm

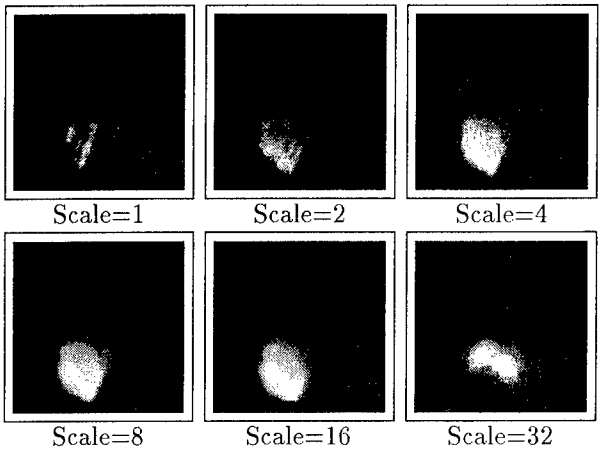


Figure 3: Flow magnitudes at scales 1,2,4,8,16 and 32 (top left to bottom right)

Figure 4 is a rescaled version of Figure 3 in which the small flow values along the arm can be more easily observed. The flow estimation along the arm at fine scales is dominated by the noise of the imaging system. As scale increases, better estimates are computed along the arm at the cost of blurring the flow of the hand. As a consequence, if motion segmentation into parts is sought, the finest scale would result in highly fragmented components, while the coarsest scale would lead to highly inaccurate boundaries for the hand.

3 A Multi-scale Flow Model

Let $I(x, y, t)$ be the image brightness at a point (x, y) at time t. The brightness constancy assumption at scale s is given by

$$I(x, y, t) = I(x + su\delta t, y + sv\delta t, t + s\delta t) \quad (1)$$

where (u, v) is the horizontal and vertical image velocity at (x, y), and δt is small. We assume, for now, that the instantaneous velocity (u, v) remains constant during the time span $s\delta t$ (leading to a displace-

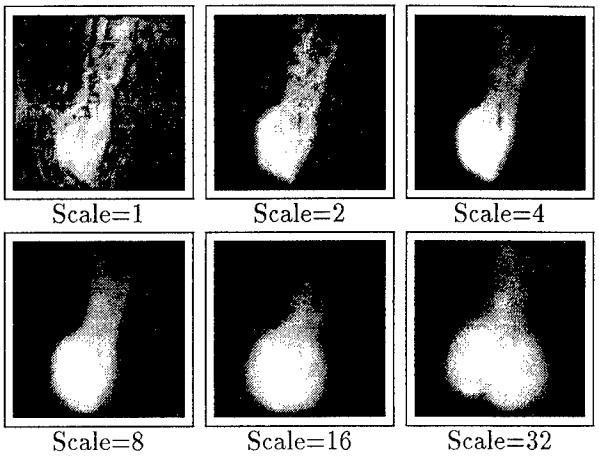


Figure 4: Enhanced flow display to show flow estimation at scales 1,2,4,8,16 and 32 (top left to bottom right)

ment $(su\delta t, sv\delta t)$). This assumption is less likely to hold with the increase of scale and can lead to violations of brightness constancy. Let the range of scales over which flow is to be estimated be $1, ..., n$. Expanding Equation (1) using a Taylor series approximation (assuming locally constant flow) and dropping terms results in

$$0 = s(I^s{}_x(x,y,t)u + I^s{}_y(x,y,t)v + I^s{}_t(x,y,t)) \quad (2)$$

where I^s is the s-th frame (forward in time relative to I) of the sequence, and $I^s{}_x, I^s{}_y$ and $I^s{}_t$ are the spatial and temporal derivatives of image I^s relative to I.

Since Equation (2) is underconstrained for computation of (u, v), it is ordinarily posed as a minimization of the least-squares error of the flow over a very small neighborhood, R, of (x, y), leading to

$$E(u,v,s) = \sum_{(x,y)\in R} (s(I^s{}_x u + I^s{}_y v + I^s{}_t))^2 \quad (3)$$

We have n equations of the form of Equation (3), one for each scale. The *scale-generalized* error is defined as

$$E_D(u,v) = \sum_{s\in 1...n} \sum_{(x,y)\in R} (s(I^s{}_x u + I^s{}_y v + I^s{}_t))^2 \quad (4)$$

Notice that Equation (4) biases the error term towards coarser scales due to the multiplication by s. Therefore, we normalize the error terms so that the minimization is in the form†

$$E_D(u,v) = \sum_{s\in 1...n} \sum_{(x,y)\in R} (I^s{}_x u + I^s{}_y v + I^s{}_t))^2 \quad (5)$$

†The same effect could have been achieved by dividing the right side of Equation (2) by s for all scales, prior to error summation.

Equation (5) gives equal weight to the error values of all scales. Since it is expected that at each point (x, y) the accuracy of instantaneous motion estimation will be scale-dependent, we introduce a weight function $W(u, v, s)$ designed (see below) to minimize the influence of the residuals of the relatively inaccurate scales. Equation (5) now becomes

$$E_D(u,v) = \sum_{s\in 1...n} \sum_{(x,y)\in R} (W(I^s{}_x u + I^s{}_y v + I^s{}_t))^2 \quad (6)$$

Instead of the least-squares minimization in Equation (6) we choose a robust estimation approach as proposed in [4], resulting in

$$E_D(u,v) = \sum_{s\in 1...n} \sum_{(x,y)\in R} \rho(W(I^s{}_x u + I^s{}_y v + I^s{}_t), \sigma_e) \quad (7)$$

where ρ is a robust error norm that is a function of a scale parameter σ_e. Since the weight function $W(u, v, s)$ should also reflect the degree of accuracy of the flow estimation we redefine it to include a scaling parameter σ_w, $W(u, v, s, \sigma_w)$. The choice of the weighting function W should satisfy the following constraints:

- It should take on values in the range $[0, ..., c]$, c typically chosen as 1.0 for computational convenience.
- For a large σ_w, W should approach 1.0 regardless of (u, v) and s.
- Given σ_w, large estimated flow (u, v) at point (x, y) should lead to higher weights for the lower scales of the error term $I^s{}_x u + I^s{}_y v + I^s{}_t$, while small flow should lead to higher weights for the higher scales.

Figure 5 reflects qualitatively the desired shape of the weighting function for a fixed σ_w. It illustrates the weighting as a function of scale s and flow magnitude $||(u, v)||$ at (x, y). The following Gaussian function satisfies the above requirements:

$$W(u,v,s,\sigma_w) = e^{-(s - \frac{n}{(\alpha||(u,v)||^2+1.0)})^2/2\sigma_w{}^2} \quad (8)$$

where $||(u, v)||^2$ is the squared magnitude of the current flow estimate at (x, y), and α is a constant. Notice that when $||(u, v)||^2 << 1.0$ the maximal weight occurs at the highest scale n, while higher values of $||(u, v)||^2$ lead to a maximal weight at lower scales; specifically, the Gaussian is centered at $\frac{n}{\alpha||(u,v)||^2+1.0}$. The scale parameter σ_w determines the width of the Gaussian, and the constants α and 1.0 can be changed to further shift the maximal weight scale location. The application of the weighting function in the estimation is as follows: In the

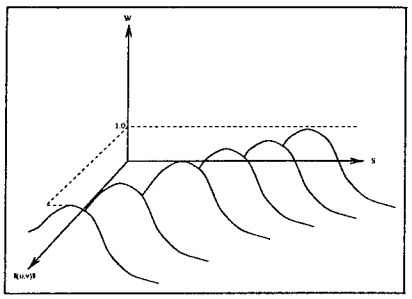


Figure 5: The weighting function as a function of s and flow magnitude $||(u, v)||$

first iteration, all scales are given equal weight (1.0) by selecting a large σ_w. Afterwards, iteratively, the estimates are refined by decreasing σ_w.

This temporal multi-scale procedure is accompanied by a spatial coarse-to-fine strategy [3] that constructs a pyramid of the spatially filtered and subsampled images (for more information see [4]) and computes the flow initially at the coarsest level and then propagates the results to finer levels. The computational aspects of the multi-scale model follow, generally, the approach proposed by Black and Anandan [4, 5].

4 Experimental Results

In the following figures we show the results of image flow computation when $\sigma_w = 20.0$ and is decreased at a rate of 0.85 for five iterations, and $\sigma_e = 100.0$ and is also decreased at a rate of 0.85 for 40 iterations. The computation is performed over 16 scales.

Figure 6 illustrates the weights at several scales during the computation of image flow (the brighter the intensity the higher the weight; weights across scales were normalized in these images to allow for comparisons). At *scale*=1 only the hand area is given high weights while the arm and the background are given very low weights. As the scale increases the weights are increased along the arm and the background while a decrease on the hand gradually takes place. At the highest scale (*scale* =16) the hand's weight is very low while the arm and the background receive high weights. Figure 7 shows the effect of the iterative refinement of the weighting function W for *scale* =1 (the finest scale) on the relative weights for different regions. The values are normalized across the five images to allow comparison. Notice that the first iteration gives high weights to the hand, and the weights given to the arm and the background are somewhat significant. The fifth iteration also gives high weights to the hand while the arm and the background have the lowest weights, and they are much lower than after the first iteration. This behavior is reversed when we consider the coarsest

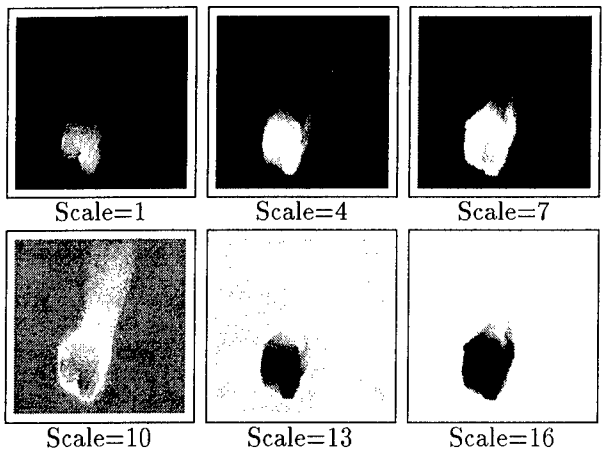


Figure 6: The weighting function W as computed at the scales 1,4,7,10,13 and 16 (top left to bottom right), expressed as an intensity image

scale, *scale* =16 (see Figure 8).

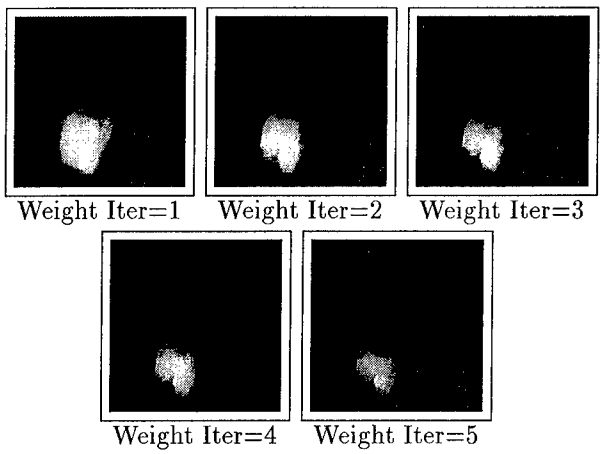


Figure 7: The weighting function W at scale 1 (finest scale) as evolved in five iterations

Figure 9 (top and middle rows) shows graphs of the individual scale flow magnitudes computed along a line drawn down the center of the arm (bottom right). These graphs correspond to the scale computations shown in Figure 3. Since the arm is *approximately* moving like a pendulum with the hand simultaneously rotating around the wrist, the flow should increase slowly along the arm and then jump considerably on the hand. This is clearly visible in these graphs. The dip in these graphs (occurring between 125 and 145) is a result of the intensity discontinuity associated with the four quadrants of the camera. Figure 9 also shows the multi-scale flow magnitude results (bottom left). The flow boundary is quite sharp. The flow magnitude along the

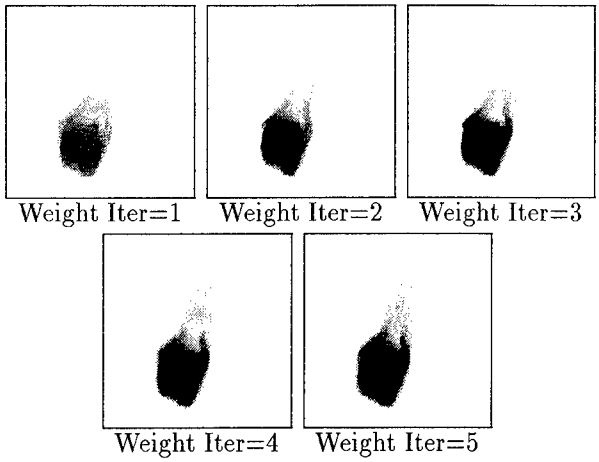


Figure 8: The weighting function W at scale 16 (coarsest scale) as evolved in five iterations

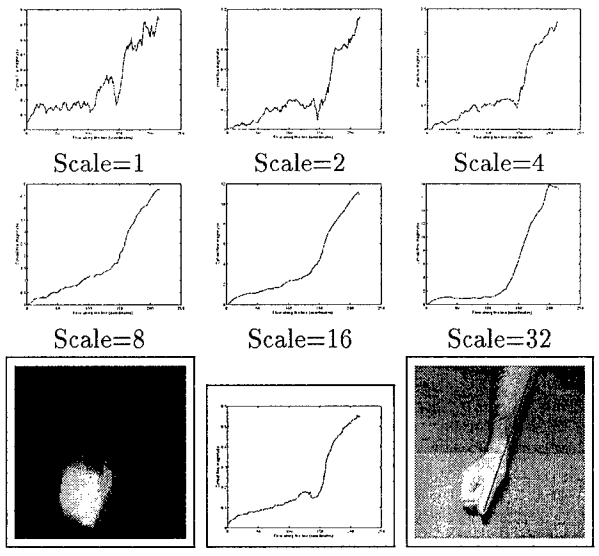


Figure 9: The flow magnitude along a line (bottom right) computed using a single scale ($s = 1, 2, 4, 8, 16$ and 32; top and middle rows), the multi-scale flow magnitudes (bottom left), and the multi-scale flow magnitudes along the line (bottom center)

line is also shown (bottom center); it reveals a very smooth change in the flow along the arm and a significant increase at the hand, with maximal flow at the finger.

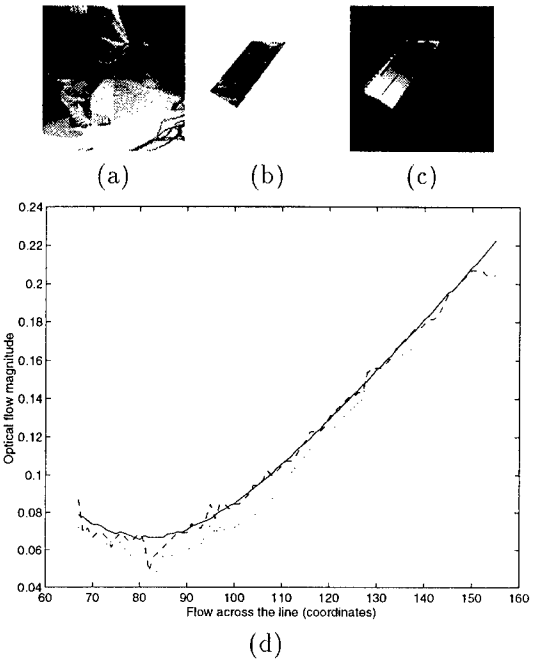


Figure 10: A synthetic motion example that compares flow magnitudes on a real image of the calf of a walking person's leg. The image (see (b)) was warped and the flow magnitudes along a line (see (c)) are shown as a solid line (see (d)). The estimates of flow magnitudes using 1 and 12 scales over the same line are shown ((d), dotted and dashed lines, respectively).

In order to compare the performance of single-scale ($scale = 1$) and multi-scale flow estimation, we generated a sequence of images using a synthetic flow model where we have ground-truth data. Figure 10 (top) shows an image of a person during a walking activity. The synthetic sequence is generated by warping the image patch of the calf forward according to a multi-scale parameterized motion model for several frames (assuming constant velocity). The estimated multi-scale (12 scales) flow magnitudes are also shown (top right). A quantitative comparison is shown, along a line on the calf, between the original flow (bottom, solid line), the single-scale flow (dotted line), and the multi-scale flow (dashed line). The multi-scale estimate is closer to the synthetic flow than the single-scale estimate. Accurate recovery of the flow is actually limited by interpolation side-effects in generating the synthetic motion.

Figure 11 shows four images taken from a long sequence of a person moving his arm around while rotating his face (consecutive images are four frames

Figure 11: Images from a sequence showing simultaneous arm and head motion

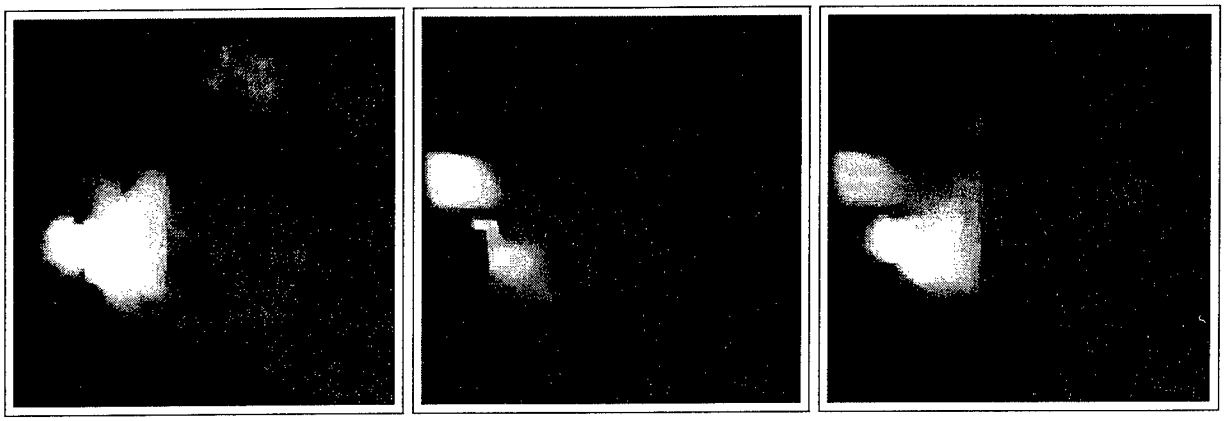


Figure 12: Horizontal, vertical and magnitude of flow from two frames (left to right). For horizontal and vertical flow, brighter value indicates greater motion leftward and upward, respectively.

Figure 13: Multi-scale ($s = 16$) horizontal, vertical and magnitude of flow (left to right). For horizontal and vertical flow, brighter value indicates greater motion leftward and upward, respectively.

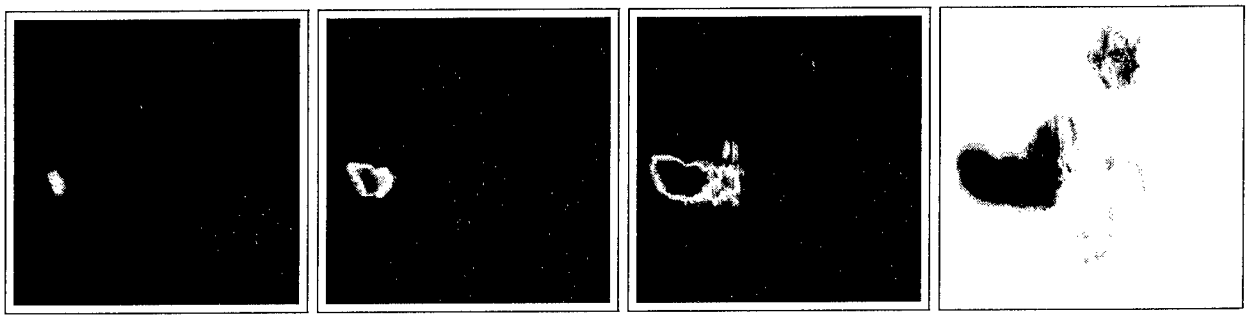


Figure 14: The weights for scales $1, 5, 10$ and 16 (dark indicates lower weight)

apart). The magnitudes of the motions of the arm and face vary significantly. Also, the flow along the arm varies, beginning with no motion at the shoulder and increasing towards the hand. In Figures 12-13 the single-scale and multi-scale flow and acceleration are shown (16 scales). The multi-scale result is more reflective of the image motion since it adaptively measures flow at the best scale while the single-scale algorithm uses parameters that lead to computing non-zero motion in the background. Figure 14 shows the weights used in computation for scales $1, 4, 8, 12$ and 16.

5 Estimation of Image Acceleration

The scale-generalized brightness constancy assumption given in Equation (1) assumes constant flow at all scales. This can be extended to include acceleration models. Let the image flow as a function of scale s be $(u(s), v(s))$. Then the brightness constancy assumption at scale s becomes

$$I(x,y,t) = I(x+\sum_s u(s)ds, y+\sum_s v(s)ds, t+s) \quad (9)$$

As a special case, if the image motion is assumed to be subject to a constant acceleration, the flow is given by

$$u(s) = x_0 + x_1 s \quad (10)$$

$$v(s) = x_2 + x_3 s \quad (11)$$

where x_1 and x_3 are the horizontal and vertical acceleration terms. Note that in the context of a long sequence this model supports a piecewise-constant acceleration assumption. If acceleration fluctuations within the scales involved in the estimation are small or fall within the performance range of the robust estimator (about 35%-40% outliers), this model holds. This flow model leads to a brightness constancy assumption of the form

$$I(x,y,t) = I(x + \sum_{i \in 1 \ldots s} (x_0 + x_1 i), y + \sum_{i \in 1 \ldots s} (x_2 + x_3 i), t + s) \quad (12)$$

Using a Taylor series expansion and dropping terms (including scale normalization), we arrive at

$$0 = I^s{}_x(x_0 + x_1 \frac{s+1}{2}) + I^s{}_y(x_2 + x_3 \frac{s+1}{2}) + I^s{}_t \quad (13)$$

The new scale-generalized error function is given by

$$E_D(u,v) = \sum_{s \in 1 \ldots n} \sum_{(x,y) \in R} \rho(W(I^s{}_x(x_0 + x_1 \frac{s+1}{2}) + I^s{}_y(x_2 + x_3 \frac{s+1}{2}) + I^s{}_t), \sigma_e) \quad (14)$$

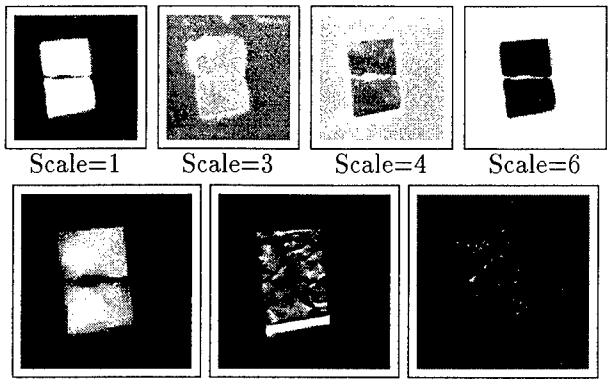


Figure 15: The weights (upper row) at scales 1, 3, 4 and 6, respectively (out of 6 scales), and the flow magnitude and vertical and horizontal accelerations (bottom row, left to right) for a falling book.

Figure 15 shows the dense flow and acceleration estimated for a falling book sequence. The top row shows the weighting function values assigned for each scale (normalized to enhance the contrast). At low scales the book region is assigned high weight while the background is assigned very low weight. This is reversed as scale is increased; at the top scale the motion of the book is so large that little weight is given to the book area. The bottom row shows the dense velocity magnitude (left) and the vertical and horizontal accelerations (center and right, respectively). Notice that the estimated horizontal acceleration is almost uniformly zero.

6 Discussion

The proposed multi-scale approach to computing optical flow and acceleration introduces explicit temporal models for image intensity and flow changes. As demonstrated here for several image sequences, a multi-scale framework can increase the accuracy of the instantaneous motion estimates and recover simultaneously both flow and acceleration.

Algorithms for motion estimation can be quite noisy since they are based on local operators applied over very small temporal neighborhoods. Temporal smoothing was proposed in [4] in a regularization framework; in contrast, our multi-scale approach employs well-understood scale-space concepts [7, 8] to create smooth estimates. Due to the integrative nature of the multi-scale estimation, motion smoothing is achieved through the estimation process.

In this paper we have developed a new multi-temporal framework for computing flow and acceleration in images. Both dense and parameterized representations were employed and demonstrations on long image sequences were provided. This approach is an extension of the popular brightness-constancy

assumption to a temporal scale-space domain. It provides for higher accuracy over a wider range of flows, and thereby provides a useful tool for the analysis of image sequences.

References

[1] S.S. Beauchemin and J.L. Barron. The Computation of Optical Flow. *ACM Computing Surveys,* Vol. 27, 1995, 433-467.

[2] J.R. Bergen, P. Anandan, K.J. Hanna and R. Hingorani. Hierarchical Model-Based Motion Estimation. In G. Sandini, editor, *ECCV-92,* LNCS Vol. 588, Springer-Verlag, 1992, 237-252.

[3] M.J. Black and P. Anandan. A Framework for Robust Estimation of Optical Flow. *ICCV,* 1993, 231-236.

[4] M.J. Black and P. Anandan. The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields. 1994 revision of Technical Report P93-00104, Xerox PARC, 1993.

[5] D.J. Fleet and A.D. Jepson. Computation of Component Image Velocity from Local Phase Information. *IJCV,* Vol. 3, 1990, 77-104.

[6] D.J. Heeger. Optical Flow Using Spatio-Temporal Filters. *IJCV,* Vol. 1, 1988, 279-302.

[7] T. Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer Academic Publishers, 1994.

[8] T. Lindeberg. A Scale Selection Principle for Estimating Image Deformations. Technical Report CVAP 196, Royal Institute of Technology, 1996.

[9] A. Singh. Incremental Estimation of Image Flow Using a Kalman Filter. *Proceedings of the IEEE Workshop on Visual Motion,* 1991, 36-43.

Understanding Object Motion

Zoran Duric[1,2], Ehud Rivlin[1,3], Azriel Rosenfeld[1]
[1]Center for Automation Research
University of Maryland
College Park, MD 20742-3275
[2]Machine Learning and Inference Laboratory
George Mason University
Fairfax, VA 22030-4444
[3]Department of Computer Science
Technion – Israel Institute of Technology
Haifa, Israel 32000

Abstract

Many types of common objects, such as tools and vehicles, usually move in simple ways when they are wielded or driven: The natural axes of the object tend to remain aligned with the local trihedron defined by the object's trajectory. The alignment can be verified by analysis of the flow field generated by the moving object; this is illustrated here for three examples, involving a wrench, a saw and a van.

1 Introduction

An object moves because it is self-propelled (e.g., a vehicle) or because it is wielded (or thrown[1]) by an agent (e.g., a tool). Motion that efficiently performs a locomotional or mechanical function requires efficient energy transfer from the vehicle's engine or the agent's arm to the object, in order to efficiently overcome the constraints imposed by the environment in which the motion takes place (air resistance, friction, etc.). Assuming that an object has natural axes (e.g the long axis of a stick), efficient force transfer requires simple relationships between the natural axes of the object and the motion trajectory. These relationships insure that the object can perform its function efficiently.

The most general model of object motion is unrestricted rigid motion. This type of motion is not common in everyday life. Usually objects are supported, and motion takes place when an object is in contact with a surface, another object, or an agent. In these cases (tool acting on a recipient object; ground vehicle) the motion becomes significantly constrained.

In our work we consider the relationship between this constrained motion and the object's geometry. To analyze this relationship we use two frames: the object frame and the frame of the motion trajectory. "Efficient" motion calls for a simple relationship between the object frame and the motion frame, and this relationship remains constant during the motion. Based on this observation we use a model called *Frenet-Serret motion* which corresponds to the motion of a moving trihedron along a space curve [8]. The parameters of the motion are given by the curvature and torsion of the space curve along which the object moves.

We use the relationship between the object frame and the motion frame to analyze image sequences. Given a sequence of images of the moving object, our analysis enables us to output the motion and trajectory parameters of the object. Knowing how the Frenet-Serret frame is changing relative to the observer gives us essential information for understanding the object's motion. Our analysis can also handle constraints on the motion. For example, the parameters of the object's trajectory depend on its speed, mass, size, and on the medium through which it moves. These factors impose bounds on the curvature and torsion of the trajectory.

In this paper we approach object motion understanding through analysis of long image sequences. A key question in this context is how to relate short-sequence motion estimation to long-sequence motion estimation. Using the Frenet-Serret frame provides

The support of the Defense Advanced Research Projects Agency and the Office of Naval Research under Grant N00014-95-1-0521 is gratefully acknowledged

[1]We assume in this paper that the propulsive force is applied to the object continuously, unlike the case of a projectile where it is applied only initially. We will not discuss projectiles further here.

us with an ability to understand motion over a long time period. We can derive the motion parameters from the parameters of the trajectory and obtain motion descriptions suitable for long sequence analysis. Using these tools we can show, for example, that rotation becomes significant only in long sequences, and that in a short sequence translation is usually dominant. We show that using simplified scene and imaging models we can get adequate local estimates (short sequence, 2-4 frames) by analyzing the images, and by observing these estimates over a long sequence we can accumulate them to describe the object's trajectory. Analysis of the trajectory parameters provides us with tools for understanding long-term object motion.

2 Related Work

Understanding object motion is based on extracting the object's motion parameters from an image sequence. Broida and Chellappa [1] proposed a framework for motion estimation of a vehicle using Kalman filtering. Weng et al. [15] assumed an object that possesses an axis of symmetry, and a constant angular momentum model which constrained the motion over a local frame subsequence to be a superposition of precession and translation. The trajectory of the center of rotation can be approximated by a vector polynomial. Changing the parameters of the model with time allows adaptation to long-term changes in the motion characteristics. Their work was based on correspondence; at least eight pairs of corresponding points were needed.

Accumulating the information obtained from the motion analysis of the sequence to achieve an estimate of the moving object's trajectory is another step toward understanding object motion. (A good survey of motion-based recognition was compiled by Cedras and Shah [5].) Bruckstein et al. [2, 3] assumed a known object model (a rigid rod or disk) and tried to recover the object's trajectory and rotation. They showed that five images are enough to recover the motion of a rod or a disk in accordance with physical laws. Techniques from algebraic geometry were used to establish the existence of solutions to the resulting polynomial equations.

Engel and Rubin [9] (and similarly Gould and Shah [10]) used motion characteristics obtained by tracking representative points on an object to identify important events corresponding to changes in direction, speed and acceleration in the object's motion.

Work has also been done on higher-level descriptions of object trajectories in terms of such concepts as stopping/starting, object interactions, and motion verbs[4, 11, 12]. This level of object motion description will not be treated in this paper.

In [6] Duric et al. tried to determine the function of an object from its motion. Given a sequence of images of a known object performing some function, they attempted to determine what that function was. They showed that the motion of an object, when combined with information about the object and its uses, provides strong constraints on the possible function being performed. Their flow-based analysis treated relatively short sequences.

In this paper a model for object trajectory analysis is used, and a constant relationship between the object frame and the motion frame is established. The use of the Frenet-Serret frame provides a vocabulary appropriate for describing long motion sequences.

3 Motion Models

3.1 Rigid Body Motion

To facilitate the derivation of the motion equations of a rigid body $\mathcal{B}$ we use two rectangular coordinate frames, one $(Oxyz)$ fixed in space, the other $(Cx_1y_1z_1)$ fixed in the body and moving with it. The coordinates X_1, Y_1, Z_1 of any point P of the body with respect to the moving frame are constant with respect to time t, while the coordinates X, Y, Z of the same point P with respect to the fixed frame are functions of t. It is assumed that these functions are differentiable with respect to t. The position of the moving frame at any instant is given by the position $\vec{d}_c = (X_c \;\; Y_c \;\; Z_c)^T$ of the origin C, and by the nine direction cosines of the axes of the moving frame with respect to the fixed frame. Let $\vec{i}$, $\vec{j}$, and $\vec{k}$ be the unit vectors in the directions of the Ox, Oy, and Oz axes, respectively; and let $\vec{i}_1$, $\vec{j}_1$, and $\vec{k}_1$ be the unit vectors in the directions of the Cx_1, Cy_1, and Cz_1 axes, respectively. For a given position $\vec{p}$ of P in $Cx_1y_1z_1$ we have the position $\vec{r}_p$ of P in $Oxyz$:

$$\begin{aligned}\vec{r}_p &\equiv \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \\ &= \begin{pmatrix} \vec{i}\cdot\vec{i}_1 & \vec{i}\cdot\vec{j}_1 & \vec{i}\cdot\vec{k}_1 \\ \vec{j}\cdot\vec{i}_1 & \vec{j}\cdot\vec{j}_1 & \vec{j}\cdot\vec{k}_1 \\ \vec{k}\cdot\vec{i}_1 & \vec{k}\cdot\vec{j}_1 & \vec{k}\cdot\vec{k}_1 \end{pmatrix}\begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix} + \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} \\ &\equiv R\vec{p} + \vec{d}_c \end{aligned} \quad (1)$$

where R is the matrix of the direction cosines (the frames are taken as right-handed so that $\det R = 1$).

If we differentiate (1) with respect to time and use the fact that $\vec{p} = R^T(\vec{r}_p - \vec{d}_c)$, we obtain

$$\dot{\vec{r}}_p = \dot{R}\vec{p} + \dot{\vec{d}}_c = \dot{R}R^T(\vec{r}_p - \vec{d}_c) + \dot{\vec{d}}_c \equiv \Omega(\vec{r}_p - \vec{d}_c) + \dot{\vec{d}}_c. \quad (2)$$

The skew matrix $\Omega = \dot{R}R^T = -R\dot{R}^T$ is the rotational velocity matrix and $\dot{\vec{d}}_c$ is the translational velocity vector. Multiplying a vector $(\vec{r}_p - \vec{d}_c)$ by the skew matrix Ω can be replaced by taking the cross product $\vec{\omega} \times (\vec{r}_p - \vec{d}_c)$ where $\vec{\omega} = (\omega_x \ \omega_y \ \omega_z)^T$ is the rotational velocity vector.

3.2 Motion along a Smooth Curve

Consider a moving frame $Cx_1y_1z_1$ (fixed in a rigid body $\mathcal{B}$), which moves with C along a space curve Γ while rotating so that the Cx_1 and Cy_1 axes concide with, respectively, the tangent and principal normal of Γ. This means that as C moves along Γ the $Cx_1y_1z_1$ frame concides with the Frenet-Serret trihedron at C: $Ctnb$. This trihedron consists of the tangent $\vec{\mathbf{t}}$, the principal normal $\vec{\mathbf{n}}$, and the binormal $\vec{\mathbf{b}}$, which are mutually orthogonal (see Figure 1). The geometry of this motion is completely defined by Γ.

Let $\vec{d}_\gamma(s)$ denote the position of C, in the fixed coordinate frame $Oxyz$, when it has moved along Γ through a total arc length of s. For any position $\vec{p}$ of a point P on $\mathcal{B}$ in $Ctnb$, the position $\vec{r}_p$ in $Oxyz$ is given by (1) with the matrix of direction cosines R suitably defined (see Figure 1). If $\vec{\mathbf{t}} = (t_1 \ t_2 \ t_3)^T$,

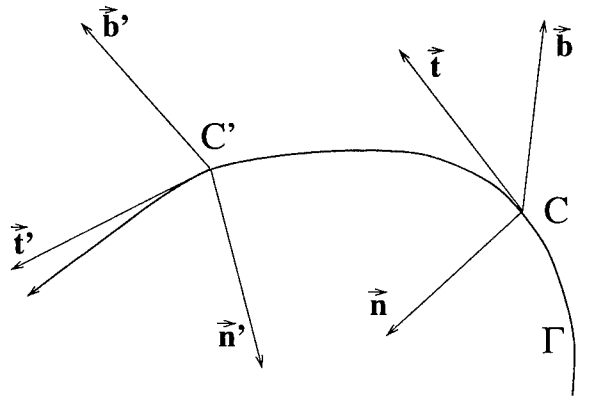


Figure 1: The Frenet-Serret coordinate frame moves along the path Γ.

$\vec{\mathbf{n}} = (n_1 \ n_2 \ n_3)^T$ and $\vec{\mathbf{b}} = (b_1 \ b_2 \ b_3)^T$ are the unit vectors along Ct, Cn and Cb, differential geometry gives us

$$\vec{\mathbf{t}} = \vec{d}_\gamma', \quad \vec{\mathbf{n}} = \kappa^{-1}\vec{d}_\gamma'', \quad \vec{\mathbf{b}} = \vec{\mathbf{t}} \times \vec{\mathbf{n}}, \quad (3)$$

where κ is the curvature of Γ. Then we have

$$R = \begin{pmatrix} t_1 & n_1 & b_1 \\ t_2 & n_2 & b_2 \\ t_3 & n_3 & b_3 \end{pmatrix}. \quad (4)$$

We have the Frenet-Serret formulas [13]

$$\vec{\mathbf{t}}' = \kappa\vec{\mathbf{n}}, \quad \vec{\mathbf{n}}' = -\kappa\vec{\mathbf{t}} + \tau\vec{\mathbf{b}}, \quad \vec{\mathbf{b}}' = -\tau\vec{\mathbf{n}} \quad (5)$$

where τ is the torsion of Γ. Using (4) and (5), (2) can be written as

$$\dot{\vec{r}}_p = \vec{\omega}_d \times (\vec{r}_p - \vec{d}_\gamma) + \vec{\mathbf{t}} \quad (6)$$

where the Darboux vector $\vec{\omega}_d = \tau\vec{\mathbf{t}} + \kappa\vec{\mathbf{b}}$ is the rotational velocity vector and the unit tangent $\vec{\mathbf{t}}$ of Γ is the translational velocity vector; the motion parameter is the arc length s. If, instead of using arc length as a motion parameter, time t is used, the rotational velocity $\vec{\omega}_d$ and translational velocity $\vec{\mathbf{t}}$ are scaled by the speed $v = ds/dt$ of the point C. In that case the equation of motion becomes

$$\dot{\vec{r}}_p = v\vec{\omega}_d \times (\vec{r}_p - \vec{d}_\gamma) + v\vec{\mathbf{t}}. \quad (7)$$

In the special case where Γ is a plane curve we have $\tau = 0$ (Γ is torsionless), and thus $\vec{\omega}_d = \kappa\vec{\mathbf{b}}$. Equation (7) then becomes

$$\dot{\vec{r}}_p = v\kappa\vec{\mathbf{b}} \times (\vec{r}_p - \vec{d}_\gamma) + v\vec{\mathbf{t}}. \quad (8)$$

3.3 Simple Motions of Objects

Objects move in reaction to forces which are being applied to them. When the forces acting on an object are added, the resultant force $\vec{F}$ determines the direction of motion and the moments of the forces (or the torques) determine the rotation of the object. If the force $\vec{F}$ is applied to the object $\mathcal{B}$ at the point P, the moment $\vec{M}$ is given by $\vec{M} = \vec{r}_p \times \vec{F}$ where $\vec{r}_p$ is the position of P relative to a point C. $\vec{M}$ has the same direction as the axis of the rotation of $\mathcal{B}$ that results from applying $\vec{F}$.

The engine of a vehicle needs to apply force to the vehicle in order to move it from one position to another. If the path is prespecified (as in the case of a ground vehicle on a road), efficient application of the force requires that the angle between the instantaneous directions of the force and the directions of the path elements be small. The force differential generates torques which help turn the vehicle around the axis of rotation normal to the (osculating) plane of the path. During a turn, the wheels rotate with different speeds; the greater the distance between the wheels the larger their difference in speed. In order

to minimize this difference the distance between the wheels needs to be small. Also, when forces are applied to the wheels the resulting torques are larger when the vehicle is moving along a short axis; but these torques need to be as small as possible to improve the handling of and minimize stresses on the vehicle. Because of all these factors the principal axis of inertia of the vehicle should be tangent to the path of the vehicle. It should be pointed out that [7] the translational velocity at any point on a ground vehicle is typically orders of magnitude larger than its rotational velocity (around the vehicle's center of mass). The rotational velocity becomes significant only when the vehicle is observed over a significant period of time (typically several frames).

In the case of a moving tool the force is used not only to move the tool, but to act on a recipient object. Therefore, the required force depends on the task. For example, sawing involves continuously exerting a force perpendicular to the path of the saw; tightening with a wrench involves continuously exerting torque around the axis of rotation. (Note that the force may not be applied to the recipient object continuously; for example, when we swing a hammer, the force is applied only when the head of the hammer hits the object.) Developing a general theory of tool motion is a subject of our continuing research.

4 Computing Motion from Image Sequences

For the purpose of estimating object motion from images we rewrite (2) in the following way:

$$\dot{\vec{r}}_p = \vec{\omega} \times (\vec{r}_p - \vec{d}_c) + \dot{\vec{d}}_c = \vec{\omega} \times \vec{r}_p + \vec{T} \qquad (9)$$

where $\vec{T} = \dot{\vec{d}}_c - \vec{\omega} \times \vec{d}_c \equiv (U \ V \ W)^T$ is the translational velocity expressed in the fixed (camera) coordinate frame $Oxyz$. We will later show how the translational velocity $\dot{\vec{d}}_c$ can be recovered from $\vec{T}$.

4.1 The Imaging Models

Let (X, Y, Z) denote the Cartesian coordinates of a scene point with respect to the fixed camera frame (see Figure 2), and let (x, y) denote the corresponding coordinates in the image plane. The equation of the image plane is $Z = f$, where f is the focal length of the camera. The perspective projection onto this plane is given by

$$x = \frac{fX}{Z}, \quad y = \frac{fY}{Z}. \qquad (10)$$

For weak perspective projection we need a reference point (X_c, Y_c, Z_c). A scene point (X, Y, Z) is first projected onto the point (X, Y, Z_c); then, through plane perspective projection, the point (X, Y, Z_c) is projected onto the image point (x, y). The projection equations are then given by

$$x = \frac{X}{Z_c} f, \quad y = \frac{Y}{Z_c} f. \qquad (11)$$

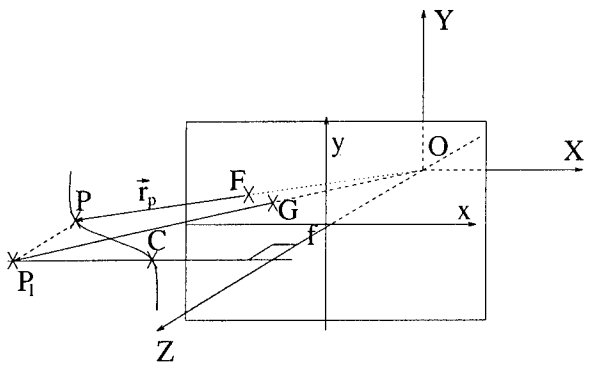


Figure 2: The plane perspective projection image of P is $F = f(X/Z, Y/Z, 1)$; the weak perspective projection image of P is obtained through the plane perspective projection of the intermediate point $P_1 = (X, Y, Z_c)$ and is given by $G = f(X/Z_c, Y/Z_c, 1)$.

4.2 The Image Motion Field and the Optical Flow Field

The instantaneous velocity of the image point (x, y) under perspective projection is obtained by taking the derivatives of (10) and using (9):

$$\begin{aligned} \dot{x} &= \frac{\dot{X}Z - X\dot{Z}}{Z^2} = \frac{Uf - xW}{Z} - \omega_x \frac{xy}{f} \\ &\quad + \omega_y \left(\frac{x^2}{f} + f\right) - \omega_z y, \qquad (12) \\ \dot{y} &= \frac{\dot{Y}Z - Y\dot{Z}}{Z^2} = \frac{Vf - yW}{Z} - \omega_x \left(\frac{y^2}{f} + f\right) \\ &\quad + \omega_y \frac{xy}{f} + \omega_z x. \qquad (13) \end{aligned}$$

Similarly, the instantaneous velocity of (x, y) under weak perspective projection is obtained by taking derivatives of (11) and using (9):

$$\begin{aligned} \dot{x} &= f\frac{\dot{X}Z_c - X\dot{Z}_c}{Z_c^2} \\ &= \frac{Uf - xW}{Z_c} + f\omega_y \frac{Z}{Z_c} - \omega_z y, \qquad (14) \\ \dot{y} &= f\frac{\dot{Y}Z_c - Y\dot{Z}_c}{Z_c^2} \\ &= \frac{Vf - yW}{Z_c} - f\omega_x \frac{Z}{Z_c} + \omega_z x. \qquad (15) \end{aligned}$$

Let $\vec{i}$ and $\vec{j}$ be the unit vectors in the x and y directions, respectively; $\dot{\vec{r}} = \dot{x}\vec{i} + \dot{y}\vec{j}$ is the projected motion field at the point $\vec{r} = x\vec{i} + y\vec{j}$. If we choose a unit direction vector $\vec{n}_r$ at the image point $\vec{r}$ and call it the normal direction, then the *normal motion field* at $\vec{r}$ is $\dot{\vec{r}}_n = (\dot{\vec{r}} \cdot \vec{n}_r)\vec{n}_r$. Here $\vec{n}_r$ can be chosen in various ways; the usual choice (as we shall now see) is the direction of the image intensity gradient.

Let $I(x, y, t)$ be the image intensity function. The time derivative of I can be written as

$$\frac{dI}{dt} = (I_x\vec{i} + I_y\vec{j}) \cdot (\dot{x}\vec{i} + \dot{y}\vec{j}) + I_t = \nabla I \cdot \dot{\vec{r}} + I_t$$

where ∇I is the image gradient and the subscripts denote partial derivatives.

If we assume $dI/dt = 0$, i.e. that the image intensity does not vary with time, then we have $\nabla I \cdot \vec{u} + I_t = 0$. The vector field $\vec{u}$ in this expression is called the *optical flow*. If we choose the normal direction $\vec{n}_r$ to be the image gradient direction, i.e. $\vec{n}_r \equiv \nabla I / \|\nabla I\|$, we then have

$$\vec{u}_n = (\vec{u} \cdot \vec{n}_r)\vec{n}_r = \frac{-I_t \nabla I}{\|\nabla I\|^2} \tag{16}$$

where $\vec{u}_n$ is called the *normal flow*.

It was shown in [14] that the magnitude of the difference between $\vec{u}_n$ and the normal motion field $\vec{r}_n$ is inversely proportional to the magnitude of the image gradient. Hence $\dot{\vec{r}}_n \approx \vec{u}_n$ when $\|\nabla I\|$ is large. Equation (16) thus provides an approximate relationship between the 3-D motion and the image derivatives. We will use this approximation later in this paper.

5 Tool Motion

We assume that the tool is (approximately) planar and that its velocity is composed of a translational velocity in the plane of the tool and a rotational velocity around an axis orthogonal to the plane of the tool.

5.1 The Image Motion Field of a Wielded Tool

Let the normal to the plane be $\vec{N} = (N_x \ N_y \ N_z)^T$; the equation of the plane orthogonal to $\vec{N}$ which passes through the point $(0, 0, Z_0)$ on the z-axis of the $Oxyz$ coordinate frame is given by

$$XN_x + YN_y + (Z - Z_0)N_z = 0. \tag{17}$$

If we assume a nondegenerate view (i.e., $N_z > 0$) for points on the plane we obtain from (17) and (10)

$$\begin{aligned} \frac{1}{Z} &= \frac{1}{fZ_0}\left(f + f\frac{XN_x}{ZN_z} + f\frac{YN_y}{ZN_z}\right) \\ &= \frac{1}{fZ_0}(f + px + qy) \end{aligned} \tag{18}$$

where $p = N_x N_z^{-1}$ and $q = N_y N_z^{-1}$. From our asumption about rotational velocity it follows that we have $\vec{\omega} = (p\omega_z \ q\omega_z \ \omega_z)$ for some ω_z.

From (12–13) and (18) we obtain the equations of projected motion for points on the plane:

$$\begin{aligned} \dot{x} &= \frac{Uf - xW}{fZ_0}(f + px + qy) - p\omega_z\frac{xy}{f} \\ &\quad + q\omega_z\left(\frac{x^2}{f} + f\right) - \omega_z y, \end{aligned} \tag{19}$$

$$\begin{aligned} \dot{y} &= \frac{Vf - yW}{fZ_0}(f + px + qy) - p\omega_z\left(\frac{y^2}{f} + f\right) \\ &\quad + q\omega_z\frac{xy}{f} + \omega_z x. \end{aligned} \tag{20}$$

Equations (19-20) relate the image (projected) motion field to the scaled translational velocity $Z_0^{-1}\vec{T} = Z_0^{-1}(U \ V \ W)^T \equiv (U_0 \ V_0 \ W_0)^T$, the rotational parameter ω_z, and the normal to the plane $(p \ q \ 1)^T$.

Let

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{pmatrix} \equiv \begin{pmatrix} n_x f \\ n_x x \\ n_x y \\ n_y f \\ n_y x \\ n_y y \\ -(n_x xy + n_y y^2)/f \\ -(n_x x^2 + n_y xy)/f \end{pmatrix} \tag{21}$$

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \end{pmatrix} \equiv \begin{pmatrix} U_0 + q\omega_z \\ U_0 p - W_0 \\ U_0 q - \omega_z \\ V_0 - p\omega_z \\ V_0 p + \omega_z \\ V_0 q - W_0 \\ W_0 q + p\omega_z \\ W_0 p - q\omega_z \end{pmatrix}. \tag{22}$$

Given the point $\vec{r} = x\vec{i} + y\vec{j}$ and the normal direction $n_x\vec{i} + n_y\vec{j}$, from (19-20) and using (21-22) the normal motion field $\dot{\vec{r}}_n \cdot \vec{n} = n_x\dot{x} + n_y\dot{y}$ is given by

$$\dot{\vec{r}}_n \cdot \vec{n} = \mathbf{a}^T\mathbf{c}. \tag{23}$$

The column vector $\mathbf{a}$ is formed of observable quantities only, while each element of the column vector $\mathbf{c}$ contains quantities which are not directly observable from the images. To estimate $\mathbf{c}$ we need estimates of $\dot{\vec{r}}_n \cdot \vec{n}$ at eight or more image points.

5.2 Estimating Tool Motion from Normal Flow

If we use the spatial image gradient as the normal direction $\vec{n}_r \equiv \nabla I/\|\nabla I\| = n_x\,\vec{i} + n_y\,\vec{j}$ and $\vec{n}_n \approx \vec{u}_n$ we can obtain an approximate equation corresponding to (23) by replacing the left hand side of (23) by normal flow $-I_t/\|\nabla I\|$. This equation involves the eight unknown elements of $\mathbf{c}$. For each point (x_i, y_i), $i = 1, \ldots, m$ of the image at which $\|\nabla I(x_i, y_i, t)\|$ is large we can write one such equation. If we have more than eight such points we have an over-determined system of equations $A\mathbf{c} \approx \mathbf{b}$; the rows of the $m \times 8$ matrix A are the vectors $\mathbf{a}_i$, and the elements of the m-vector $\mathbf{b}$ are $-(\partial I(x_i, y_i, t)/\partial t)/\|\nabla I(x_i, y_i, t)\|$.

We seek the solution of the system for which $\|\mathbf{b} - A\mathbf{c}\|$ is minimal. This solution is the same as the solution of the system $A^T A\mathbf{c} = A^T\mathbf{b}$. We use linear least squares to obtain the parameter vector $\mathbf{c}$.

Since we have assumed that the translation is in the plane of the tool we have $\vec{N} \cdot \vec{T} = 0$, or equivalently

$$(p\ q\ 1)^T \cdot (U_0\ V_0\ W_0)^T = U_0 p + V_0 q + W_0 = 0. \quad (24)$$

After estimating $\mathbf{c}$ we can use (22) and (24) to obtain $U_0, V_0, W_0, \omega_z, p$, and q. From (22) we have

$$U_0 p + V_0 q - 2W_0 = c_2 + c_6;$$

using (22) and (24) we obtain

$$\begin{aligned} W_0 &= -\frac{c_2 + c_6}{3}, \quad U_0 p = \frac{2c_2 - c_6}{3}, \\ V_0 q &= \frac{2c_6 - c_2}{3}. \end{aligned} \quad (25)$$

From (22) and (25) we obtain

$$\begin{aligned} c_7 &= W_0 q + p\omega_z = -\frac{c_2 + c_6}{3} q + p(U_0 q - c_3) \\ &= -c_3 p + \frac{c_2 - 2c_6}{3} q \end{aligned} \quad (26)$$

Similarly we have

$$\begin{aligned} c_8 &= W_0 p - q\omega_z = -\frac{c_2 + c_6}{3} p - q(c_5 - V_0 p) \\ &= \frac{c_6 - 2c_2}{3} p - c_5 q \end{aligned} \quad (27)$$

From (26-27) we obtain p and q and by substituting their values into (22) and (25) we obtain U_0, V_0, and ω_z. Finally, we obtain

$$\vec{N} = (p\ q\ 1)^T (1 + p^2 + q^2)^{-\frac{1}{2}}$$

and

$$\|\vec{\omega}\| = \sqrt{\omega_z^2 + p^2\omega_z^2 + q^2\omega_z^2}.$$

We have estimated the translational velocity $\vec{T}$ and the rotational velocity $\vec{\omega}$ in the camera coordinate system $Oxyz$. We are interested in the translational and the rotational velocity expressed in the Frenet-Serret frame $Otnb$. By comparing equations (2), (8) and (9) we obtain

$$\vec{\omega} = v\kappa\vec{\mathbf{b}}, \quad \vec{\mathbf{b}} = \vec{N}\mathrm{sgn}\omega_z, \quad v\kappa = \|\vec{\omega}\| \quad (28)$$

where sgn stands for the 'sign of' function. Also, from (2), (8) and (9) we have

$$\begin{aligned} (U_0\ V_0\ W_0)^T &= Z_0^{-1}\vec{T} = Z_0^{-1}(\dot{\vec{d}}_c - \vec{\omega} \times \vec{d}_c) \\ &= Z_0^{-1}(v\vec{\mathbf{t}} - \vec{\omega} \times \vec{d}_\gamma) \end{aligned}$$

and thus

$$\frac{v\vec{\mathbf{t}}}{Z_0} = (U_0\ V_0\ W_0)^T + \frac{\vec{\omega} \times \vec{d}_\gamma}{Z_0}. \quad (29)$$

Note that in equation (29) the quantities Z_0 and $\vec{d}_\gamma$ (the position of the point C, the origin of the $Otnb$ frame) are not known. However, let $\vec{d}_\gamma = (X_c\ Y_c\ Z_c)^T$ be the position of C and let (x_c, y_c) be the image of C (either the tip or the center of mass of the tool). From (18) we obtain

$$\frac{fZ_0}{Z_c} = f + px_c + qy_c$$

so that (29) can be written as

$$\begin{aligned} \frac{v\vec{\mathbf{t}}}{Z_0} &= (U_0\ V_0\ W_0)^T + \frac{Z_c}{fZ_0}\vec{\omega} \times \frac{f(X_c\ Y_c\ Z_c)^T}{Z_c} \\ &= (U_0\ V_0\ W_0)^T + \frac{\vec{\omega} \times (x_c\ y_c\ f)^T}{f + px_c + qy_c}. \end{aligned} \quad (30)$$

From (30) we obtain the unit vector in the tangent direction $\vec{\mathbf{t}}$ by normalizing $v\vec{\mathbf{t}}/Z_0$. Finally, we obtain the unit vector in the normal direction using

$$\vec{\mathbf{n}} = \vec{\mathbf{b}} \times \vec{\mathbf{t}}. \quad (31)$$

Equations (28), (30) and (31) define the Frenet-Serret frame $Otnb$ expressed in the camera coordinate system. Equation (28) gives us the curvature κ up to an unknown factor v (linear velocity). We conclude that the Frenet-Serret motion can be recovered up to the speed v; note that the translational velocity $v\vec{\mathbf{t}}/Z_0$ does not help here because of the unknown depth Z_0.

Finally, we need to recover the orientation of the tool coordinate frame (its long and short axes) in the $Otnb$ frame. We find the long and the short axes of the tool as the principal axes of the set of tool points. The long axis l of the tool and the origin O of the fixed (camera) coordinate frame $Oxyz$ define a

plane Π_l. Since the image l' of l lies in this plane we can find Pi_l using l' in place of l. Because we have assumed a nondegenerate view we have two cases: (i) if the tangent vector $\vec{\mathbf{t}}$ lies in Π_l the motion is along l; (ii) if the normal vector $\vec{\mathbf{n}}$ lies in Π_l the motion is orthogonal to l.

We check if the vector lies in the plane Π_l using the following simple algorithm. Let $\vec{p}_1 = (x_1 \ y_1 \ f)^T$ and $\vec{p}_2 = (x_2 \ y_2 \ f)^T$ be the positions of two endpoints on the image l' of l. The normal $\vec{N}_\Pi$ of the plane Π_l is given by

$$\vec{N}_\Pi = \vec{p}_1 \times \vec{p}_2.$$

If the vector $\vec{\mathbf{t}}$ lies in the plane Π_l we have $\vec{N}_\Pi \times \vec{\mathbf{t}} \approx 0$. So to find the relative orientation of the tool frame and the *Otnb* frame we only need to find which one of the inner products $|\vec{N}_\Pi \cdot \vec{\mathbf{t}}|$ and $|\vec{N}_\Pi \cdot \vec{\mathbf{n}}|$ is smaller. (Note that while one of the vectors $\vec{\mathbf{t}}$ and $\vec{\mathbf{n}}$ lies in the plane Π_l the other vector is not always orthogonal to Π_l .)

6 Vehicle Motion

We assume that the motion of the vehicle is planar and that it has a small rotational velocity around the axis orthogonal to the plane of motion. The translational velocity is dominant and at any time t the motion can be approximated by pure translational motion.

6.1 The Image Motion Field of a Moving Vehicle

From (14–15) we obtain the (approximate) equations of projected motion for points on a vehicle under weak perspective:

$$\dot{x} = \frac{Uf - xW}{Z_c}, \tag{32}$$

$$\dot{y} = \frac{Vf - yW}{Z_c}. \tag{33}$$

Equations (32-33) relate the image (projected) motion field to the scaled translational velocity $Z_c^{-1}\vec{T} = Z_c^{-1}(U \ V \ W)^T$.

Let

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \equiv \begin{pmatrix} n_x f \\ n_y f \\ -n_x x - n_y y \end{pmatrix}, \tag{34}$$

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \equiv \begin{pmatrix} UZ_c^{-1} \\ VZ_c^{-1} \\ WZ_c^{-1} \end{pmatrix}. \tag{35}$$

Given the point $\vec{r} = x\vec{\imath} + y\vec{\jmath}$ and the normal direction $n_x\vec{\imath} + n_y\vec{\jmath}$, from (32-33) and using (34-35) the normal motion field $\dot{\vec{r}}_n \cdot \vec{n} = n_x\dot{x} + n_y\dot{y}$ is given by $\dot{\vec{r}}_n \cdot \vec{n} = \mathbf{a}^T\mathbf{c}$. The column vector $\mathbf{a}$ is formed of observable quantities only, while each element of the column vector $\mathbf{c}$ contains quantities which are not directly observable from the images. To estimate $\mathbf{c}$ we need estimates of $\dot{\vec{r}}_n \cdot \vec{n}$ at three or more image points.

6.2 Estimating Vehicle Motion from Normal Flow

As in Section 5.2 we use linear least squares to estimate parameter vector $\mathbf{c}$ from the normal flow.

In the case of a moving vehicle the parameters of interest are the vehicle's trajectory and its *rate of approach*. The rate of approach

$$\nu = \frac{W}{Z_c}$$

(measured in $\sec^{-1}$) is equivalent to the inverse of the *time to collision* and corresponds to the rate with which an object is approaching the camera (or receding from it). The rate $\nu = 0.1/\text{sec}$ means that every second the object travels 0.1 of the distance between the observer and its current position. A negative rate of approach means that the object is going away from the camera.

The direction of motion c gives us the tangent vector $\vec{\mathbf{t}} = c/||c||$ of the Frenet-Serret frame. If the direction of motion changes over time we can use the Frenet-Serret formulas (5) to recover the (scaled) curvature $v\kappa$ of the trajectory. Given the tangent direction $\vec{\mathbf{t}}_0$ at time t and the tangent direction $\vec{\mathbf{t}}_1$ at time $t + \Delta t$ we have

$$\vec{\mathbf{n}}_0 = v\kappa\vec{\mathbf{n}} \approx \frac{\vec{\mathbf{t}}_1 - \vec{\mathbf{t}}_0}{\Delta t}. \tag{36}$$

The unit vector in the direction $\vec{\mathbf{n}}_0$ at time t is the normal vector of the *Otnb* frame and the scaled curvature is given by $v\kappa = ||\vec{\mathbf{n}}_0||$. Finally, we obtain

$$\vec{\mathbf{b}} = \vec{\mathbf{t}} \times \vec{\mathbf{n}}. \tag{37}$$

Equations (36) and (37) give us the normal $\vec{\mathbf{b}}$ to plane of motion and the rotational velocity of turning (yaw) $\vec{\omega} = v\kappa\vec{\mathbf{b}}$.

7 Experiments

7.1 Motions of Tools

We tested our motion analysis algorithm under full perspective on two image sequences of tools in motion. The first sequence, two frames of which are

shown in Figure 3, was a 200-image sequence of the movement of a wrench tightening a bolt.

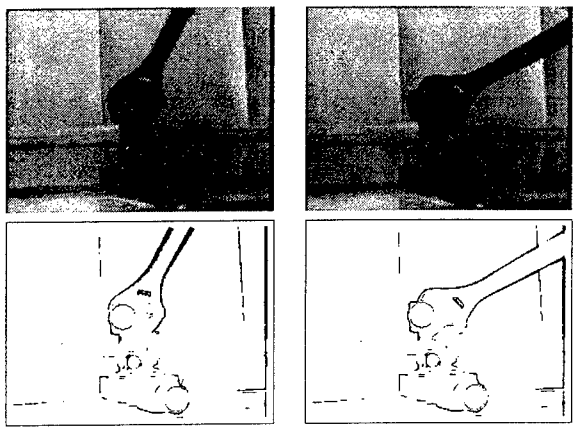


Figure 3: An experiment using a wrench: frames 30 and 100. Top images: the input images. Bottom images: results of flow computation.

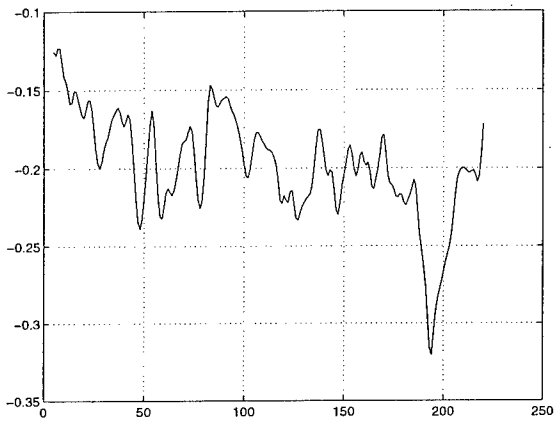


Figure 4: Results of experiments on the wrench sequence. The graph shows rotational velocity in radians/sec.

The motion of the wrench was a rotation (to turn the bolt) around an axis approximately orthogonal to the plane of the image. The rotational velocity is shown in Figure 4; it is given in radians/sec and it corresponds to the scaled curvature $v\kappa$. Figure 5 shows the orientation of the principal axis of the wrench and the instantaneous translational velocity vector of its centroid (obtained using equation (30)), both measured in radians. As we see, the translational velocity vector remains approximately orthogonal to the principal axis throughout the motion sequence. The Frenet-Serret frame has its binormal $\vec{\mathbf{b}}$ in the direction of the negative of the z-axis, its tangent $\vec{\mathbf{t}}$ in the image plane and orthogonal to the principal axis of the wrench, and its normal $\vec{\mathbf{n}}$ in the

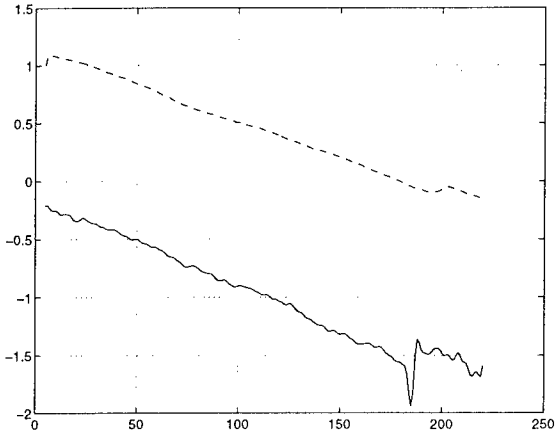


Figure 5: Results of experiments on the wrench sequence. The solid line corresponds to the orientation (in radians) of the instantaneous direction of translation of the centroid of the wrench, and the dashed line corresponds to the orientation (in radians) of the principal axis of the wrench.

image plane and oriented from the centroid of the wrench toward the bolt.

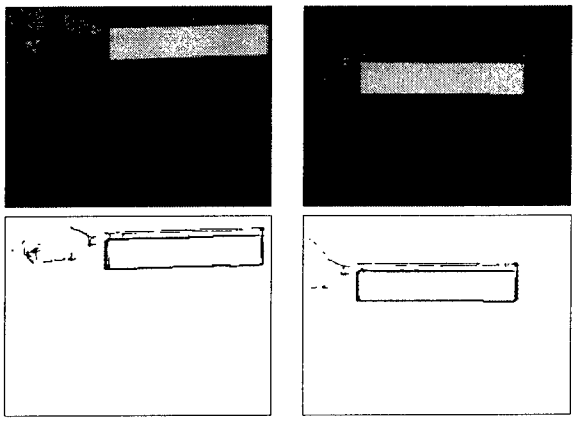


Figure 6: An experiment using a saw: frames 30 and 100. Top images: the input images. Bottom images: results of flow computation.

We also tested our motion analysis algorithm on a 200-image sequence of a saw doing a periodic motion. Figure 6 presents part of the sequence. Flow results are given below each image. The motion of the saw was pure translation ($\|\vec{\omega}\| = 0$). As can be seen from Figure 7 the motion is mostly frontoparallel (the z component of the translational velocity is small). As Figure 8 shows, the motion is periodic in the direction of the principal axis of inertia. It is a simple case of a (periodic) straight line motion with the Frenet-Serret frame corresponding to the principal axes of the saw; $\vec{\mathbf{t}}$ corresponds to the longest axis, and $\vec{\mathbf{b}}$ to the shortest axis.

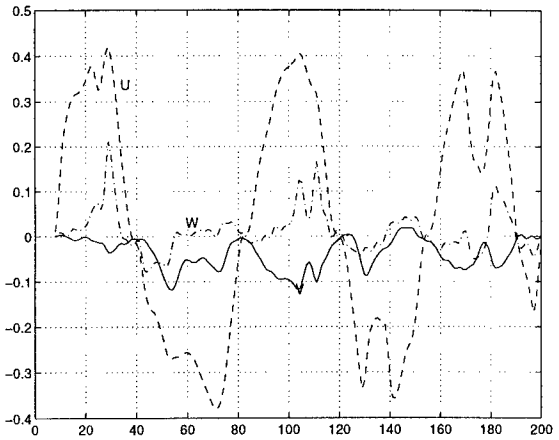


Figure 7: Results of experiments on the saw sequence. U, V, W are the scaled (by an unknown distance Z_0^{-1}) components of the relative translational velocity.

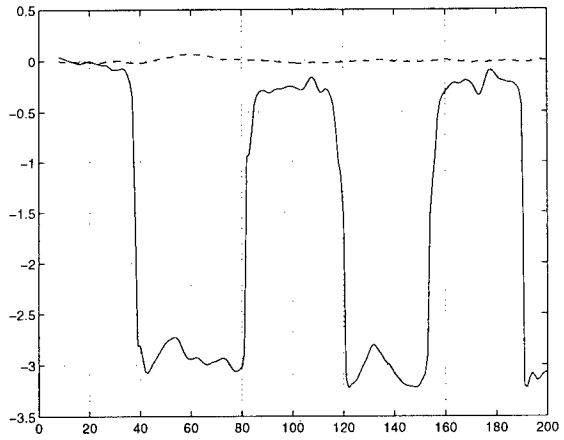


Figure 8: Results of experiments on the saw sequence. The solid line corresponds to the orientation (in radians) of the instantaneous direction of motion of the saw, and the dashed line corresponds to the orientation (in radians) of the principal axis of the saw.

7.2 Motions of Vehicles

In this experiment we used an image sequence of a van taken from another vehicle following the van, and we used the algorithms for weak perspective. The sequence consisted of 56 frames (slightly less than two seconds). Figure 9 shows frames 5, 25, and 45 as well as the corresponding normal flow on the van. Figure 10 shows estimated values of UZ_c^{-1}, VZ_c^{-1}, and WZ_c^{-1}. These values correspond to the relative translation of the van and the vehicle carrying the camera (observer coordinate system). Because of our choice of the coordinate system the rate of approach ν corresponds to the negative of WZ_c^{-1}, i.e. $\nu = -WZ_c^{-1}$. The graph shows that there is an impending collision (rate of approach greater than 1 sec^{-1}). Around the 20th frame the rate of approach becomes zero (as do all the velocity components) and after that it becomes negative because the van starts pulling away from the vehicle carrying the camera.

These graphs show that the motion components have a simple behavior; before they reach their extremal values they can be approximated by straight lines, indicating constant relative accelerations.

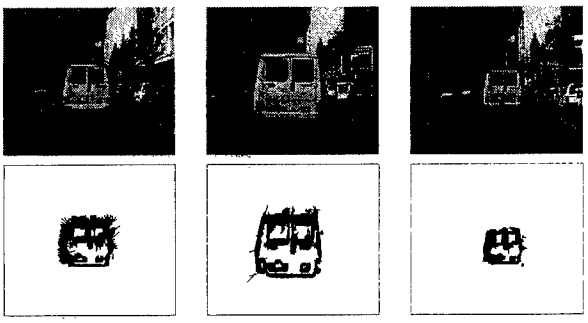


Figure 9: Frames 5, 25, and 45 of the van sequence. The normal flow results are shown below the corresponding image frames.

8 Conclusions

Many types of common objects, such as tools and vehicles, usually move in simple ways when they are wielded or driven: The natural axes of the object tend to remain aligned with the local trihedron defined by the object's trajectory. In this paper we have considered the relationship between this constrained motion and the object's geometry. To analyze this relationship we have used two frames: the object frame and the frame of the motion trajectory. Assuming a constant relationship between the object frame and the motion frame during the motion, we have used *Frenet-Serret motion* as a motion model. Using the Frenet-Serret frame has provided us with an ability to understand motion over a long time period.

We have derived equations for describing the motions of tools and vehicles under full and weak perspective. We have recovered descriptions of an object's motion and the space curve along which the object moves, using relatively long image sequences. The motion and trajectory parameters provide a low-level description for understanding the motions of vehicles. For understanding tools in motion one needs additional knowledge about the tool and the context. This is a direction for further research.

It is the need for efficient force transfer that imposes

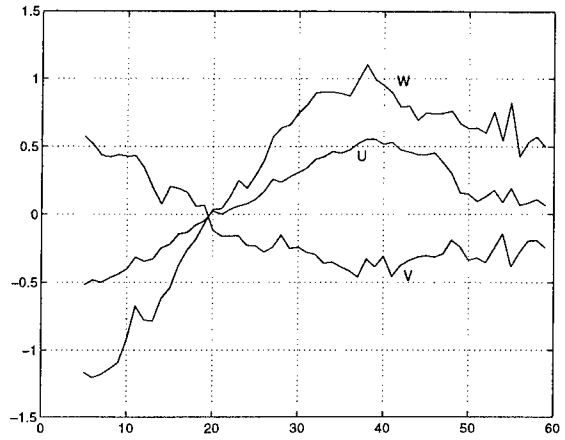


Figure 10: Results of experiments on the van sequence. U, V, W are the scaled (by an unknown distance Z_c^{-1}) components of the relative translational velocity.

a simple and constant relationship between the natural axes of the object and the motion trajectory. We have used this functional constraint in analyzing the motions of tools and ground vehicles. Expanding this analysis to other classes of objects (e.g. air vehicles) is another direction for future research.

References

[1] T. J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:90–99, 1986.

[2] A. M. Bruckstein, R. J. Holt, and A. N. Netravali. How to catch a crook. *Journal of Visual Communication and Image Representation*, 5:273–281, 1994.

[3] A. M. Bruckstein, R. J. Holt, and A. N. Netravali. How to track a flying saucer. *Journal of Visual Communication and Image Representation*, 7:196–204, 1996.

[4] H. Buxton and R. Howarth. Watching behaviour: The role of context and learning. In *Proc. International Conference on Image Processing*, volume 2, pages 797–800, 1996.

[5] C. Cedras and M. Shah. Motion-based recognition: A survey. *Image and Vision Computing*, 13:129–155, 1995.

[6] Z. Duric, J. Fayman, and E. Rivlin. Function from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:105–122, 1995.

[7] Z. Duric and A. Rosenfeld. Image sequence stabilization in real time. *Real-Time Imaging*, 2:271–284, 1996.

[8] Z. Duric, A. Rosenfeld, and L. S. Davis. Egomotion analysis based on the Frenet-Serret motion model. *International Journal of Computer Vision*, 15:105–122, 1995.

[9] S. Engel and J. Rubin. Detecting visual motion boundaries. In *Proc. Workshop on Motion*, pages 107–111, 1986.

[10] K. Gould and M. Shah. The trajectory primal sketch: A multi-scale scheme for representing motion characteristics. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 79–85, 1989.

[11] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14:609–615, 1996.

[12] D. Koller, H. Heinze, and H.H. Nagel. Algorithmic characterization of vehicle trajectories from image sequences by motion verbs. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 90–95, 1991.

[13] E. Kreyszig. *Differential Geometry*. University of Toronto Press, Toronto, Canada, 1959.

[14] A. Verri and T. Poggio. Against quantitative optical flow. In *Proc. International Conference on Computer Vision*, pages 171–180, 1987.

[15] J. Weng, T.S. Huang, and N. Ahuja. 3-D motion estimation, understanding, and prediction from noisy image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:370–389, 1987.

Representing local motion as a probability distribution matrix and object tracking

Yoav Rosenberg

Institute of Computer Science
The Hebrew University
Jerusalem Israel 91904
yoavr@cs.huji.ac.il

Michael Werman

Institute of Computer Science
The Hebrew University
Jerusalem Israel 91904
werman@cs.huji.ac.il
http://www.cs.huji.ac.il/~werman/

Abstract

Due to the aperture problem, local motion between two images can be precisely computed only at corners while at other points only partial information is available. Therefore the motion is often represented as a 2-D Gaussian random variable. The motivation for using this representation is that the Kalman filter and other methods can easily be used. However, as we show in this paper, the Gaussian approximation is valid only in special cases and often causes severe loss of data. As an alternative, we introduce a method to extract and represent displacement as a probability distribution matrix, and introduce a filter and other tools that works directly on these matrices. Using these tools, we implement a 2-D real-time tracking system. A more detailed version of this paper can be found in [5].

1 Introduction

Computing local motion is an essential stage in many computer vision tasks, such as object tracking, image registration and structure from motion. In many cases, the algorithm tracks a set of points between two or more frames. However, an exact tracking of the point's location is possible only when the point lies on a corner. In other cases, only partial information can be extracted. The prevalent solution is to represent the displacement of the feature point as a 2-D random Gaussian variable, where the covariance matrix contains the directional edge information[8][1][3]. The advantage of this representation is its simplicity and the possibility to use efficient tools such as the Kalman filter.

In [8] a Gaussian representation and a Kalman filter is used to compute an optical flow map in a video sequence. In [6] [2], Kalman filter is used for obtaining structure from motion. In [1] a 3-D tracking system is implemented using spatial and temporal filters.

In many cases, this Gaussian assumption is not valid, and leads to errors and information loss. Consider Fig. 1, two points are given and the motion between two frames is to be found. We take a small window around each point and compute the correlation between the two frames. A and B in the first row of Fig. 1 are the correlation results and in the second row, their Gaussian approximations are shown. It can be seen that the Gaussian approximation does not represent the distribution, especially in B. A consequence of trying to impose this approximation is that if the two points belongs to one scene moving uniformly and we want to find the probability distribution matrix of the whole motion using these two measurement, we should superimpose the probability matrices and normalize them. The product matrix is shown in C. It can be shown that although each point does not contains enough information about the scene's motion, we get a better idea about it from the product matrix. Trying the same with the Gaussian approximations gives the matrix C in the second row. The result is a Gaussian with a mean value of $(2, 1)$ which is far from the true motion of $(0, 0)$. This result simulates what happens when we use the Kalman filter. Thus, we can conclude that using Gaussian approximations and the Kalman filter is in many cases erroneous. Generally speaking, the Gaussian approximation is reasonable only when

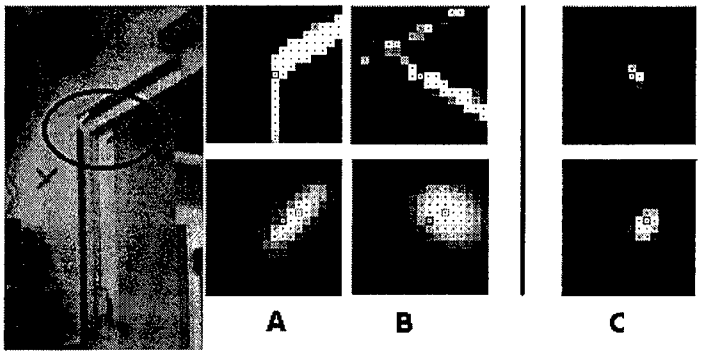


Figure 1: The probability matrices of points that cannot be approximated as Gaussians. First row: The matrices of the points within the circle (a, b). The product matrix (c). Second row: Gaussian approximation

the probability matrix is symmetric.

Another problem with Gaussian approximations concerns point tracking. When a point is not a corner, there is not enough information in the point's matrix to compute the point's displacement unambiguously. When approximating the matrix with a Gaussian, the mean is taken as the point's motion. But looking at Fig. 1,B one can see that the mean value is not a reasonable choice, because it does not coincide with a likely region of the matrix. Using this value as the point's displacement will cause the point to miss the tracked feature.

These examples demonstrate why the Gaussian approximation is not always valid. An alternative method suggested in this paper, is to represent the point's local motion as a probability distribution matrix. This will be demonstrated in a 2-D tracking system.

Tracking a set of points from an object can be used in order to track the whole object. Most methods aggregate the points motion by first making a decision about each single point, this is problematic as the single point's motion cannot usually be computed reliably without using information from all the other tracked points. Using the matrix representation, we have implemented a motion detection and tracking system which overcome these problems. The system works in real-time with live video input, even when the camera rotates. In this paper we will not describe the whole system due to space limitations. The system is used to demonstrate the matrix representation with the additional tools. This will be done by describing the main module in the tracking system.

Section 2 introduces the probability matrix formulation of a displacement and shows how a displacement's confidence can be computed from this matrix. Section 3 shows how temporal filtering can be computed directly from these matrices. Section 4 describes the computation of the object's mean motion directly from the matrices and how to compute each point's motion using this mean motion. Section 5 shows experimental results. A more detailed version of this paper can be found in [5].

2 Displacement as a probability distribution matrix

As we have shown, the displacement of a tracked point cannot be represented as a particular single value or as a Gaussian variable. The method we introduce is based on cross-SSD (sum of squared differences). The displacement is represented as a discrete probability distribution where no assumption is made about the shape of the distribution.

Formulation of the problem: Two images $\psi_1(i,j)$ and $\psi_2(i,j)$ are taken from a video sequence of a dynamic scene. Given a point $\mathbf{p}$ in ψ_1, let $\mathcal{W}$ be a window surrounding this point, and assume that all the pixels inside $\mathcal{W}$ have the same displacement as $\mathbf{p}$. We want to compute the probability distribution of the displacement of $\mathbf{p}$ between the two frames.

Let the displacement be $\mathbf{d} = (u,v)$ and let $\mathcal{W}_1$ be the window around $\mathbf{p}$ in ψ_1 and $\mathcal{W}_2$ be the window around $\mathbf{p}+\mathbf{d}$ in ψ_2. Let $P(\mathcal{W}_2 \mid \mathcal{W}_1, \mathbf{d})$ be a known function of the probability distribution of $\mathcal{W}_2$ given $\mathbf{d}$ and $\mathcal{W}_1$.

In many cases it can be assumed that the possible values for $\mathbf{d} = (u,v)$ are within the range: $U_{min}..U_{max}$, $V_{min}..V_{max}$. Let Y be defined as

$$\mathbf{Y}_{u,v} = P(\mathcal{W}_2 \mid \mathcal{W}_1, \mathbf{d} = (u,v))$$

Using Bayes' law:

$$P(\mathbf{d} \mid \mathcal{W}_1, \mathcal{W}_2) = \frac{P(\mathcal{W}_2 \mid \mathcal{W}_1, \mathbf{d})P(\mathbf{d})}{P(\mathcal{W}_2)} \qquad (1)$$

where:

$$P(\mathcal{W}_2) = \sum_{\mathbf{d}} P(\mathcal{W}_2 \mid \mathcal{W}_1, \mathbf{d})P(\mathbf{d})$$

After substituting $\mathbf{Y}_{u,v} = P(\mathcal{W}_2 \mid \mathcal{W}_1, \mathbf{d} = (u,v))$ we get:

$$P(\mathbf{d} = (u,v) \mid \mathcal{W}_1, \mathcal{W}_2) = \frac{\mathbf{Y}_{u,v}\ P(\mathbf{d} = (u,v))}{\sum_{(u,v)} \mathbf{Y}_{u,v}\ P(\mathbf{d} = (u,v))} \qquad (2)$$

$P(\mathbf{d} = (u,v))$ is the apriori probability that the displacement is $\mathbf{d}$. If no prior information is available, we take $P(\mathbf{d})$ to be constant.

It is still necessary to compute $P(\mathcal{W}_2 \mid \mathcal{W}_1, \mathbf{d})$. Let us assume that the displacement of $\mathbf{p}$ is known to be $\mathbf{d} = (u,v)$. Given the window $\mathcal{W}_1$ in the first image, the match between it and $\mathcal{W}_2$ is not perfect because of noise. The noise is generated by several sources: the camera noise, rotations, quantization errors etc. The probability distribution of the overall expected noise is very hard to compute, we present a simple method to compute the probability matrix, but other methods can work as well.

Given the displacement $\mathbf{d} = (u,v)$, the sum of squared differences of $\mathcal{W}$ between the two windows is:

$$SSD(\mathbf{d}) = \sum_{i,j \in \mathcal{W}_1} (\psi_2(i + \mathbf{d}_x, j + \mathbf{d}_y) - \psi_1(i,j))^2$$

We model the distribution of $\mathcal{W}_2$ as a function of the SSD:

$$P(\mathcal{W}_2|\mathbf{d}) = f(SSD(\mathbf{d}), \sigma^2)$$

Assuming that the only factor that can be measured or estimated is the mean SSD value σ^2, the Maximum Entropy Criteria gives:

$$P(\mathcal{W}_2|\mathbf{d}) = c\ \exp(-SSD(\mathbf{d})/\sigma^2)$$

where c is a normalization factor.

To conclude, we have introduced a method of representing the displacement as a probability distribution matrix using SSD measurements. An example

for the probability matrices is depicted in Fig. 1. The lower row shows the Gaussian approximation of the distribution, and as can be seen, this approximation eliminates almost all the available information of the matrix.

If the motion also contains rotations, this will be interpreted as more system noise, and the probability distribution will be less sharp. However, if the rotation is not too large, the matrix still contains enough information to be useful. The effect of rotation is smaller as the window size is smaller.

3 Using temporal filtering in tracking a point

Many systems use temporal filtering to improve point tracking. When the local motion is represented as a Gaussian, the Kalman filter can be utilized. However, the Kalman filter cannot handle the probability matrix representation described above. In this section we represent an alternative filter implementation that use the probability matrix P as an input. The technique implemented here can be formalized as a filter which is a generalization of the Kalman filter for any distribution [4].

Let the **process X** be a moving point where the state $\mathbf{x}_t$ is the point's 2-D velocity vector at time t. The probability distribution of $\mathbf{x}_t$ is the matrix $\mathbf{P}_{\mathbf{x}_t}$ where each entry (u, v) is the probability that the point's velocity is $\mathbf{d} = (u, v)$. This is a square matrix with size $2\mathbf{d}_{max} + 1$. Assume for the moment that the point's motion has a constant velocity. At each time interval t, a new frame ψ_2 from the video sequence is available. We refer to it as the **measurement**. From the new frame, the matrix $\mathbf{Y}(u, v) = P(\mathcal{W}_2 \mid \mathbf{d} = (u, v))$ is computed. (see Section 2).

Using Eq. 2, the posteriori probability distribution of the process given the measurement is:

$$P^{+}_{\mathbf{x}_t}(u, v) = \frac{P^{-}_{\mathbf{x}_t}(u, v)\ P(\mathcal{W}_2 \mid \mathbf{d} = (u, v))}{P(\mathcal{W}_2)} \qquad (3)$$

where $+$ stand for posteriori distribution, and $-$ for the apriori.

Equation 3 is equivalent to computing the matrix $P^{-}_{\mathbf{x}_t}(u, v)\ P(\mathcal{W}_2 \mid \mathbf{d} = (u, v))$ and normalizing the matrix to give $\sum\sum P^{+}_{\mathbf{x}_t}(u, v) = 1$.

This simple procedure is the temporal filtering step. The problem is that it is very restrictive to assume that every point moves with a constant velocity. A better assumption is that: a) the point moves with roughly a constant velocity, and b) that this is true only for short time intervals.

The process noise. The assumption of a roughly constant velocity can be interpreted as noise in the process, that is: $\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{n}$ where n is noise.

$$\mathbf{P}^{-}_{\mathbf{x}_t} = \mathbf{P}_{\mathbf{x}_{t-1}} \otimes P_{\mathbf{n}}$$

where $P_{\mathbf{n}}$ is the noise distribution matrix.

The filter adaptivity. Equation 3 gives equal weight for all the measurements. As it is assumed, the constant velocity assumption is valid only for short time intervals, so the filter must be adaptive. The adaptivity can be achieved by giving newer measurement more weight than older ones. An accepted method is to use exponential weighting where a measurement with an age Δt ($\Delta t = 1, 2, 3, ...$) is given a weight of $w(\Delta t) = \lambda^{\Delta t}$ [7]
Given the process current probability matrix $\mathbf{P}_{\mathbf{x}_t}$, This weighting can be implemented as follows. For each element (i, j) in $\mathbf{P}_{\mathbf{x}_t}$:

$$P'_{\mathbf{x}_t}(i, j) = P_{\mathbf{x}_t}(i, j)^{1/\lambda}$$

and normalize the matrix. λ determines the filter memory length. For example, a value of $\lambda = 1.1$ causes the seventh previous measurement to have half the weight as the new measurement. This way, the filter 'forgets' previous measurements and becomes adaptive.

An example of the filter implementation can be seen in Fig. 3. In the left column are the probability matrices before filtering, and in the middle column - after filtering. The temporal filtering sharpens the probability distribution and enhances the information about the point's motion.

4 Combining information from several points to implement 2D object tracking

Object tracking is implemented by randomly choosing N points on the object and tracking them. We do not assume that the points are on edges or corners. Besides, the object can be non-rigid and can have any motion in 3-D space. The tracking scheme implemented here is 2-D tracking, where the object's motion is defined as the **mean** translation of the N tracked points on it.

A simple but problematic way to implement such a tracking, is first to decide the motion of each of the tracked points, and then to calculate the mean motion. The problem is that the exact motion of each tracked point cannot usually be decided directly from its matrix. This is possible only for corner points, but we do not want to limit ourselves to such points.

Our method is as follows: in the first stage, the object's mean motion is calculated directly from the tracked points' filtered probability matrices. In second stage, the translation of each point is calculated using its probability matrix, and using the calculated mean object motion. (Section 4.1). In the third stage, a correction is made for each point's location in order to prevent drift. (Sections 4.2).

Let $\mathbf{P}_i$ be the distribution matrix of the i-th tracked point. Define the matrix $\mathbf{P}_{sum}$ as the sum of the distribution matrices of the N points, where the (x, y)

entry of this matrix has the value:

$$\mathbf{P}_{sum}(x,y) = \sum_{i=1}^{N} \mathbf{P}_i(x,y)$$

For each value (u,v), the expected squared error is:

$$MSE(u,v) = \frac{\sum_x \sum_y P_{sum}(x,y)[(x-u)^2 + (y-v)^2]}{\sum_x \sum_y P_{sum}(x,y)}$$

$(\bar{u},\bar{v})$ minimizes MSE, where:

$$\bar{u} = \frac{\sum_{x,y} P_{sum}(x,y)\ x}{\sum_{x,y} P_{sum}(x,y)},\ \bar{v} = \frac{\sum_{x,y} P_{sum}(x,y)\ y}{\sum_{x,y} P_{sum}(x,y)}$$

Choose $(\bar{u},\bar{v})$ as the object's motion.

Using this method it is possible to track an object by tracking some points on it even though the points are not on corners or edges. Next we will see how we can use the total motion information to compute the motion of each single point.

4.1 Updating the tracked point's location

The object's mean motion is computed directly from the probability distribution matrices of the tracked points. However, in order to track the points to the next frame, the point's displacement needs to be computed. This can be done locally only for corner points. In our tracking system we choose the motion of every point so that it is as close as possible to the object's mean motion.

The method is first to find the set of entries for the matrix $\mathbf{P}_{\mathbf{x}_t}$ that satisfy:

$$S = \{\mathbf{s} \in S \mid P_{\mathbf{x}_t}(\mathbf{s})/P_{\mathbf{x}_t\ max} \geq \epsilon\} \qquad (4)$$

and from this set choose the entry $\mathbf{s} = (u,v)$ with minimal size of $(u-\bar{u})^2 + (v-\bar{v})^2$ as the displacement $\mathbf{d}$.

4.2 The drift problem

With the method described here the local motion is computed up to one pixel accuracy. This can cause the tracked points to drift from its initial position after a few frames. In order to prevent drift we need a computation method whose *mean* computation error over a long sequence of frames is always zero, but can still have instantaneous errors. In [5] we show how to achieve such an accuracy using entropy measurements on the point's distribution matrices.

5 Experimental results

We have implemented the tracking system described in the above sections. Before the tracking, twenty points were randomly chosen on the tracked object. Notice that most of the points are not on edges, and therefore are more difficult to track. The tracking was carried out for forty frames. The video was taken with a moving camera, so that both the object and the background are moving. In Fig. 2, eight out of the forty frames are depicted. It can be seen that most of the points where successfully tracked through the sequence.

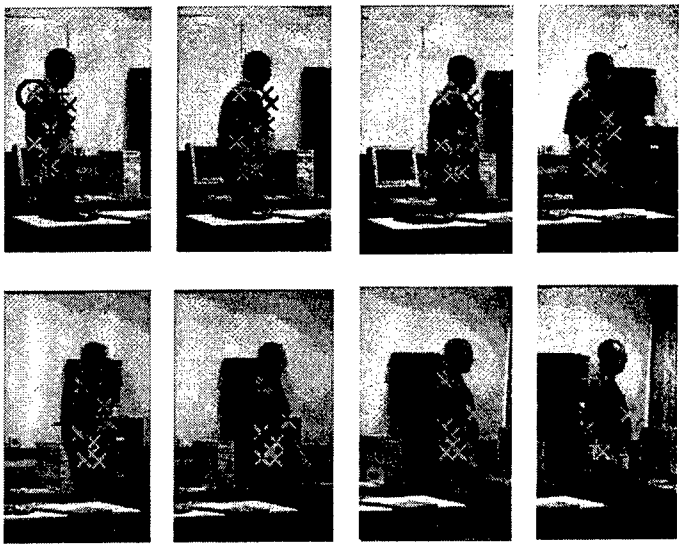


Figure 2: Tracking a moving person with a moving camera. Eight frames are shown out of forty. Points marked as X detect motion. The big cross is the points' center of mass.

In the figure, points marked as X are those where motion was detected, and the points marked as $+$, are where no motion was detected. The motion detector is implemented very simply by checking the probability of the entry $(0,0)$ in the process's probability matrix $\mathbf{P}_{\mathbf{x}_t}$.

The process's noise distribution matrix $P_{\mathbf{n}}$ was chosen as $\mathcal{N}(0,0.75)$. The adaptivity factor is $k = 0.7$, The matrices size are $(-8..8,-8..8)$ so that translation up to 8 pixels between frames can be detected. the window size was 5×5 pixels. In Fig. 3, the probability matrices are depicted. The left column is a sequence of measurement matrices $\mathbf{Y}$ belonging to the tracked point marked with a circle in Fig. 2. It can be seen that the distributions are not always similar to a Gaussian. The middle column is the probability matrix of the process $\mathbf{P}_{\mathbf{x}_t}$, i.e. the motion distribution after filtering. The filter effect can be seen as the probabilities are sharper than in the instantaneous measurement matrix.

In the right column, the sequence of the sum matrices $\mathbf{P}_{sum}(x,y)$, which represents the motion of all the points is shown. The matrices' entries with a point inside represents entries with probability close to maximum, i.e. the entries belongs the set S. The motion of each tracked point is chosen from this set as the one closest to the object's mean motion, as discussed in 4.1.

This example demonstrates how by using the tools developed in this paper, a real-time tracking can be implemented relatively easily. Tracking is achieved

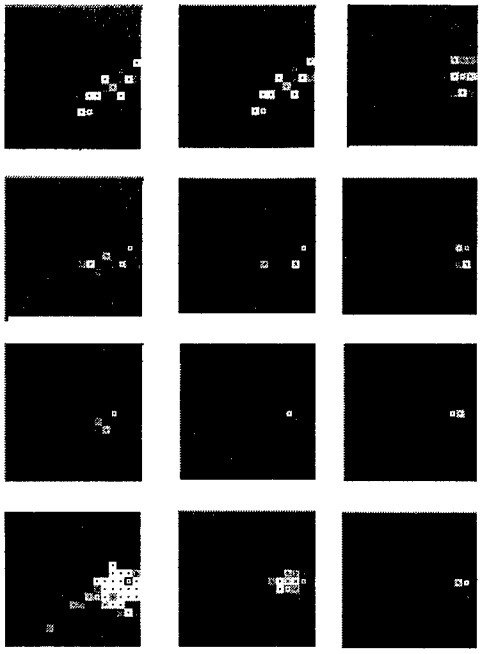


Figure 3: The probability matrices of the first four frames. The current measurement between the last two frames (left), after temporal filtering (middle) and the sum matrix from all the tracked points (right).

with only a small number of tracked points, without feature detection. It works even when the object cannot be considered rigid, (for example - the hand motion which is different than the body motion), and when the object has rotation. The tracking works well with live video on a PC in real-time.

For comparison, we have tried to repeat the tracking when we approximate the matrices with Gaussians and using the Kalman filter. The results are described in [5].

6 Summary

It has been shown that local motion cannot be represented as a Gaussian and that a better approach is to represent it with a distribution matrix. A filter and other tools for manipulating these matrices are derived. A 2-D tracking system was implemented using these tools. The distribution matrix representation allowed us to implement the tracking of the whole object without forcing us to make a prior decisions about single points which made the system much more robust.

We believe that the matrix representation can be used to solve other computer-vision tasks such as optical flow, image registration, structure from motion and 3-D tracking.

References

[1] D. B. Gennery. Visual tracking of known three-dimensional objects. *International Journal of Computer Vision*, pages 243–270, 1992.

[2] R. Szeliski L. Matthies, T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *IJCV*, (3):209–236, 1989.

[3] I.D. Reid and D.W. Murray. Tracking foveated corner clusters using affine structure. *International Conf. on Computer Vision*, pages 76–83, 1993.

[4] Y. Rosenberg and M. Werman. A general disribution filter for measurements with any probability distribution. *CVPR 1997*, page http://www.cs.huji.ac.il/papers/IP/CVPR97/index.html.

[5] Y. Rosenberg and M. Werman. Representing local motion as a probability distribution matrix for object tracking and other applications. *http://www.cs.huji.ac.il/papers/IP/CVPR97/index.html.*

[6] A. Shmuel and M. Werman. Active vision: 3d from an image sequence. *Tenth International Conference on Pattern Recognition*, A:48–54, 1990.

[7] A. Singh. An estimation-theoretic framework for image-flow computation. *International Conf. on Computer Vision*, pages 168–177, 1990.

[8] A Singh. Incremental estimation of image-flow using a kalman filter. *Workshop on Visual Motion*, pages 36–43, 1991.

Moving Object Detection and Event Recognition Algorithms for Smart Cameras

Thomas J. Olson
Frank Z. Brill

Texas Instruments
Research & Development
P.O. Box 655303, MS 8374, Dallas, TX 75265
E-mail: olson@csc.ti.com, brill@ti.com
http://www.ti.com/research/docs/iuba/index.html

Abstract

Smart video cameras analyze the video stream and translate it into a description of the scene in terms of objects, object motions, and events. This paper describes a set of algorithms for the core computations needed to build smart cameras. Together these algorithms make up the Autonomous Video Surveillance (AVS) system, a general-purpose framework for moving object detection and event recognition. Moving objects are detected using change detection, and are tracked using first-order prediction and nearest neighbor matching. Events are recognized by applying predicates to the graph formed by linking corresponding objects in successive frames.The AVS algorithms have been used to create several novel video surveillance applications. These include a video surveillance shell that allows a human to monitor the outputs of multiple cameras, a system that takes a single high-quality snapshot of every person who enters its field of view, and a system that learns the structure of the monitored environment by watching humans move around in the scene.

1 Introduction

Video cameras today produce images, which must be examined by humans in order to be useful. Future 'smart' video cameras will produce information, including descriptions of the environment they are monitoring and the events taking place in it. The information they produce may include images and video clips, but these will be carefully selected to maximize their useful information content. The symbolic information and images from smart cameras will be filtered by programs that extract data relevant to particular tasks. This filtering process will enable a single human to monitor hundreds or thousands of video streams.

The research described in this report was sponsored in part by the DARPA Image Understanding Program.

In pursuit of our research objectives [Flinchbaugh, 1997], we are developing the technology needed to make smart cameras a reality. Two fundamental capabilities are needed. The first is the ability to describe scenes in terms of object motions and interactions. The second is the ability to recognize important events that occur in the scene, and to pick out those that are relevant to the current task. These capabilities make it possible to develop a variety of novel and useful video surveillance applications.

1.1 Video Surveillance and Monitoring Scenarios

Our work is motivated by a several types of video surveillance and monitoring scenarios.

Indoor Surveillance: Indoor surveillance provides information about areas such as building lobbies, hallways, and offices. Monitoring tasks in lobbies and hallways include detection of people depositing things (e.g., unattended luggage in an airport lounge), removing things (e.g., theft), or loitering. Office monitoring tasks typically require information about people's identities: in an office, for example, the office owner may do anything at any

time, but other people should not open desk drawers or operate the computer unless the owner is present. Cleaning staff may come in at night to vacuum and empty trash cans, but should not handle objects on the desk.

Outdoor Surveillance: Outdoor surveillance includes tasks such as monitoring a site perimeter for intrusion or threats from vehicles (e.g., car bombs). In military applications, video surveillance can function as a sentry or forward observer, e.g. by notifying commanders when enemy soldiers emerge from a wooded area or cross a road.

In order for smart cameras to be practical for real-world tasks, the algorithms they use must be robust. Current commercial video surveillance systems have a high false alarm rate [Ringler and Hoover, 1995], which renders them useless for most applications. For this reason, our research stresses robustness and quantification of detection and false alarm rates. Smart camera algorithms must also run effectively on low-cost platforms, so that they can be implemented in small, low-power packages and can be used in large numbers. Studying algorithms that can run in near real time makes it practical to conduct extensive evaluation and testing of systems, and may enable worthwhile near-term applications as well as contributing to long-term research goals.

1.2 Approach

The first step in processing a video stream for surveillance purposes is to identify the important objects in the scene. In this paper it is assumed that the important objects are those that move independently. Camera parameters are assumed to be fixed. This allows the use of simple change detection to identify moving objects. Where use of moving cameras is necessary, stabilization hardware and stabilized moving object detection algorithms can be used (e.g. [Burt et al, 1989, Nelson, 1991]. The use of criteria other than motion (e.g., salience based on shape or color, or more general object recognition) is compatible with our approach, but these criteria are not used in our current applications.

Our event recognition algorithms are based on graph matching. Moving objects in the image are tracked over time. Observations of an object in successive video frames are linked to form a directed graph (the *motion graph*). Events are defined in terms of predicates on the motion graph. For instance, the beginning of a chain of successive observations of an object is defined to be an ENTER event. Event detection is described in more detail below.

Our approach to video surveillance stresses 2D, image-based algorithms and simple, low-level object representations that can be extracted reliably from the video sequence. This emphasis yields a high level of robustness and low computational cost. Object recognition and other detailed analyses are used only after the system has determined that the objects in question are interesting and merit further investigation.

1.3 Research Strategy

The primary technical goal of this research is to develop general-purpose algorithms for moving object detection and event recognition. These algorithms comprise the Autonomous Video Surveillance (AVS) system, a modular framework for building video surveillance applications. AVS is designed to be updated to incorporate better core algorithms or to tune the processing to specific domains as our research progresses.

In order to evaluate the AVS core algorithms and event recognition and tracking framework, we use them to develop applications motivated by the surveillance scenarios described above. The applications are small-scale implementations of future smart camera systems. They are designed for long-term operation, and are evaluated by allowing them to run for long periods (hours or days) and analyzing their output.

The remainder of this paper is organized as follows. The next section discusses related work. Section 3 presents the core moving object detection and event recognition algorithms, and the mechanism used to establish the 3D positions of objects. Section 4 presents applications that have been built using the AVS framework. The final section discusses the current state of the system and our future plans.

2 Related Work

Our overall approach to video surveillance has been influenced by interest in selective attention and task-oriented processing [Swain and Stricker, 1991, Rimey and Brown, 1993, Camus et al., 1993]. The fundamental problem with current video surveillance technology is that the useful information density of the images delivered to a human is very low; the vast majority of surveillance video frames contain no useful information at all. The fundamental role of the smart camera described above is to reduce the volume of data produced by the camera, and increase the value of that data. It does this by discarding irrelevant frames, and by expressing the information in the relevant frames primarily in symbolic form.

2.1 Moving Object Detection

Most algorithms for moving object detection using fixed cameras work by comparing incoming video frames to a reference image, and attributing significant differences either to motion or to noise. The algorithms differ in the form of the comparison operator they use, and in the way in which the reference image is maintained. Simple intensity differencing followed by thresholding is widely used [Jain et al., 1979, Yalamanchili et al., 1982, Kelly et al., 1995, Bobick and Davis, 1996, Courtney, 1997] because it is computationally inexpensive and works quite well in many indoor environments. Some algorithms provide a means of adapting the reference image over time, in order to track slow changes in lighting conditions and/or changes in the environment [Karmann and von Brandt, 1990, Makarov, 1996a]. Some also filter the image to reduce or remove low spatial frequency content, which again makes the detector less sensitive to lighting changes [Makarov et al., 1996b, Koller et al., 1994].

Recent work [Pentland, 1996, Kahn et al., 1996] has extended the basic change detection paradigm by replacing the reference image with a statistical model of the background. The comparison operator becomes a statistical test that estimates the probability that the observed pixel value belongs to the background.

Our baseline change detection algorithm uses thresholded absolute differencing, since this works well for our indoor surveillance scenarios. For applications where lighting change is a problem, we use the adaptive reference frame algorithm of Karmann and von Brandt [1990]. We are also experimenting with a probabilistic change detector similar to Pfinder [Pentland, 1996.

Our work assumes fixed cameras. When the camera is not fixed, simple change detection cannot be used because of background motion. One approach to this problem is to treat the scene as a collection of independently moving objects, and to detect and ignore the visual motion due to camera motion [e.g. Burt et al., 1989] Other researchers have proposed ways of detecting features of the optical flow that are inconsistent with a hypothesis of self motion [Nelson, 1991].

In many of our applications moving object detection is a prelude to person detection. There has been significant recent progress in the development of algorithms to locate and track humans. Pfinder (cited above) uses a coarse statistical model of human body geometry and motion to estimate the likelihood that a given pixel is part of a human. Several researchers have described methods of tracking human body and limb movements [Gavrila and Davis, 1996, Kakadiaris and Metaxas, 1996] and locating faces in images [Sung and Poggio, 1994, Rowley et al., 1996]. Intille and Bobick [1995] describe methods of tracking humans through episodes of mutual occlusion in a highly structured environment. We do not currently make use of these techniques in live experiments because of their computational cost. However, we expect that this type of analysis will eventually be an important part of smart camera processing.

2.2 Event Recognition

Most work on event recognition has focussed on events that consist of a well-defined sequence of primitive motions. This class of events can be converted into spatiotemporal patterns and recognized using statistical pattern matching techniques. A number of researchers have demonstrated algorithms for recognizing gestures and sign language [e.g., Starner and Pentland, 1995]. Bobick and Davis [1996] describe a method of recognizing ste-

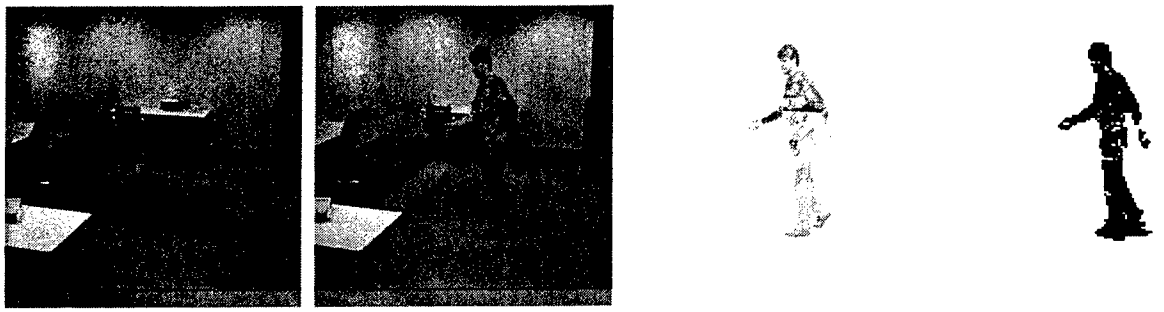


Figure 1: Image processing steps for moving object detection.

reotypical motion patterns corresponding to actions such as sitting down, walking, or waving.

Our approach to event recognition is based on the video database indexing work of Courtney [1997], which introduced the use of predicates on the motion graph to represent events. Motion graphs are well suited to representing abstract, generic events such as 'depositing an object' or 'coming to rest', which are difficult to capture using the pattern-based approaches referred to above. On the other hand, pattern-based approaches can represent complex motions such as 'throwing an object' or 'waving', which would be difficult to express using motion graphs. It is likely that both pattern-based and abstract event recognition techniques will be needed to handle the full range of events that are of interest in surveillance applications.

3 AVS Tracking and Event Recognition Algorithms

This section describes the core technologies that provide the video surveillance and monitoring capabilities of the AVS system. There are three key technologies: moving object detection, visual tracking, and event recognition. The moving object detection routines determine when one or more objects enter a monitored scene, decide which pixels in a given video frame correspond to the moving objects versus which pixels correspond to the background, and form a simple representation of the object's image in the video frame. This representation is referred to as a *motion region*, and it exists in a single video frame, as distinguished from the *world objects* which exist in the world and give rise to the motion regions.

Visual tracking consists of determining correspondences between the motion regions over a sequence of video frames, and maintaining a single representation, or *track*, for the world object which gave rise to the sequence of motion regions in the sequence of frames. Finally, event recognition is a means of analyzing the collection of tracks in order to identify events of interest involving the world objects represented by the tracks.

3.1 Moving Object Detection

The moving object detection technology we employ is a 2D change detection technique similar to that described in Jain et al. [1979] and Yalamanchili et al [1982]. Prior to activation of the monitoring system, an image of the background, i.e., an image of the scene which contains no moving or otherwise interesting objects, is captured to serve as the *reference image*. When the system is in operation, the absolute difference of the current video frame from the reference image is computed to produce a *difference image*. The difference image is then thresholded at an appropriate value to obtain a binary image in which the "off" pixels represent background pixels, and the "on" pixels represent "moving object" pixels. The four-connected components of moving object pixels in the thresholded image are the motion regions (see Figure 1).

Simple application of the object detection procedure outlined above results in a number of errors, largely due to the limitations of thresholding. If the threshold used is too low, camera noise and shadows will produce spurious objects; whereas if the threshold is too high, some portions of the objects in the scene will fail to be separated from the back-

ground, resulting in *breakup*, in which a single world object gives rise to several motion regions within a single frame. Our general approach is to allow breakup, but use grouping heuristics to merge multiple connected components into a single motion region and maintain a one-to-one correspondence between motion regions and world objects within each frame.

One grouping technique we employ is 2D morphological dilation of the motion regions. This enables the system to merge connected components separated by a few pixels, but using this technique to span large gaps results in a severe performance degradation. Moreover, dilation in the image space may result in incorrectly merging distant objects which are nearby in the image (a few pixels), but are in fact separated by a large distance in the world (a few feet).

If 3D information is available, the connected component grouping algorithm makes use of an estimate of the size (in world coordinates) of the objects in the image. The bounding boxes of the connected components are expanded vertically and horizontally by a distance measured in feet (rather than pixels), and connected components with overlapping bounding boxes are merged into a single motion region. The technique for estimating the size of the objects in the image is described in section 3.4 below.

3.2 Tracking

The function of the AVS tracking routine is to establish correspondences between the motion regions in the current frame and those in the previous frame. We use the technique of Courtney [1997], which proceeds as follows. First assume that we have computed 2D velocity estimates for the motion regions in the previous frame. These velocity estimates, together with the locations of the centroids in the previous frame, are used to project the locations of the centroids of the motion regions into the current frame. Then, a *mutual nearest-neighbor* criterion is used to establish correspondences.

Let P be the set of motion region centroid locations in the previous frame, with p_i one such location. Let p'_i be the projected location of p_i in the current frame, and let be the set of all such projected locations in the current frame. Let C be the set of motion region centroid locations in the current frame. If the distance between p'_i and $c_i \in C$ is the smallest for all elements of C, *and* this distance is also the smallest of the distances between c_i and all elements of P' (i.e., p'_i and c_i are mutual nearest neighbors), then establish a correspondence between p_i and c_i by creating a bidirectional *strong link* between them. Use the difference in time and space between p_i and c_i to determine a velocity estimate for c_i, expressed in pixels per second. If there is an existing track containing p_i, add c_i to it. Otherwise, establish a new track, and add both p_i and c_i to it.

The strong links form the basis of the tracks with a high-confidence of their correctness. Video objects which do not have mutual nearest neighbors in the adjacent frame may fail to form correspondences because the underlying world object is involved in an event (e.g., enter, exit, deposit, remove). In order to assist in the identification of these events, objects without strong links are given unidirectional *weak links* to the their (non-mutual) nearest neighbors. The weak links represent potential ambiguity in the tracking process.The motion regions in all of the frames, together with their strong and weak links, form a *motion graph.*

Figure 2 depicts a sample motion graph. In the figure, each frame is one-dimensional, and is represented by a vertical line (F0 - F18). Circles represent objects in the scene, the dark arrows represent strong links, and the gray arrows represent weak links. An object enters the scene in frame F1, and then moves through the scene until frame F4, where it deposits a second object. The first object continues to move through the scene, and exits at frame F6. The deposited object remains stationary. At frame F8 another object enters the scene, temporarily occludes the stationary object at frame F10 (or is occluded by it), and then proceeds to move past the stationary object. This second moving object reverses directions around frames F13 and F14, returns to remove the stationary object in frame F16, and finally exits in frame F17. An additional object enters in frame F5 and exits in frame F8 without interacting with any other object.

As indicated by the striped fill patterns in Figure 2, the correct correspondences for the tracks are am-

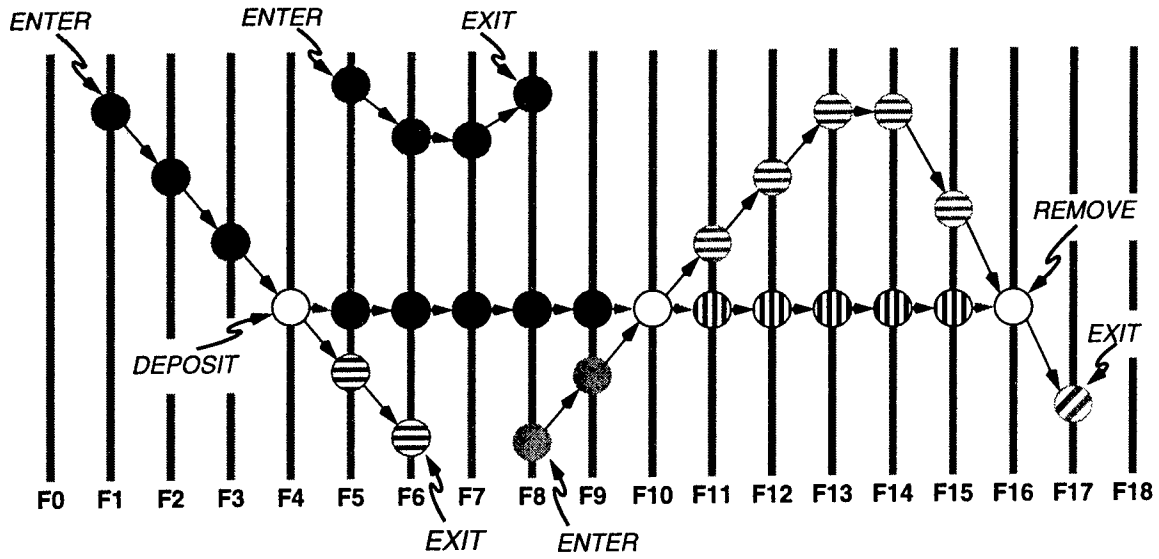


Figure 2: Event detection in the motion graph.

biguous after object interactions such as the occlusion in frame F10. The AVS system resolves this ambiguity where possible by preferring to match moving objects with moving objects, and stationary objects with stationary objects. The distinction between moving and stationary tracks is computed using thresholds on the velocity estimates, and hysteresis for stabilizing transitions between moving and stationary.

Following an occlusion (which may last for several frames) the frames immediately before and after the occlusion are compared (e.g., frames F9 and F11 in Figure 2). The AVS system examines each stationary object in the pre-occlusion frame, and searches for its correspondent in the post-occlusion frame (which should be exactly where it was before, since the object is stationary). This procedure resolves a large portion of the tracking ambiguities. General resolution of ambiguities resulting from multiple moving objects in the scene is a topic for further research. The AVS system may benefit from inclusion of a "closed world tracking" facility such as that described by Intille and Bobick [1995a, 1995b].

3.3 Event Recognition

Certain features of tracks and pairs of tracks correspond to events. For example, the beginning of a track corresponds to an ENTER event, and the end corresponds to an EXIT event. In an on-line event detection system, it is preferable to detect the event as near in time as possible to the actual occurrence of the event. The previous system which used motion graphs for event detection [Courtney, 1997] operated in a batch mode, and required multiple passes over the motion graph, precluding on-line operation. The AVS system detects events in a single pass over the motion graph, as the graph is created. However, in order to reduce errors due to noise, the AVS system introduces a slight delay of n frame times (n=3 in the current implementation) before reporting certain events. For example, in Figure 2, an enter event occurs on frame F1. The AVS system requires the track to be maintained for n frames before reporting the enter event. If the track not maintained for the required number of frames, it is ignored, and the enter event is not reported, e.g., if $n > 4$, the object in Figure 2 which enters in frame F5 and exits in frame F8 will not generate any events.

A track that splits into two tracks, one of which is moving, and the other of which is stationary, corresponds to a DEPOSIT event. If a moving track intersects a stationary track, and then continues to move, but the stationary track ends at the intersection, this corresponds to a REMOVE event. The remove event can be generated as soon as the remover disoccludes the location of the stationary object which was removed, and the system can determine that the stationary object is no longer at that location.

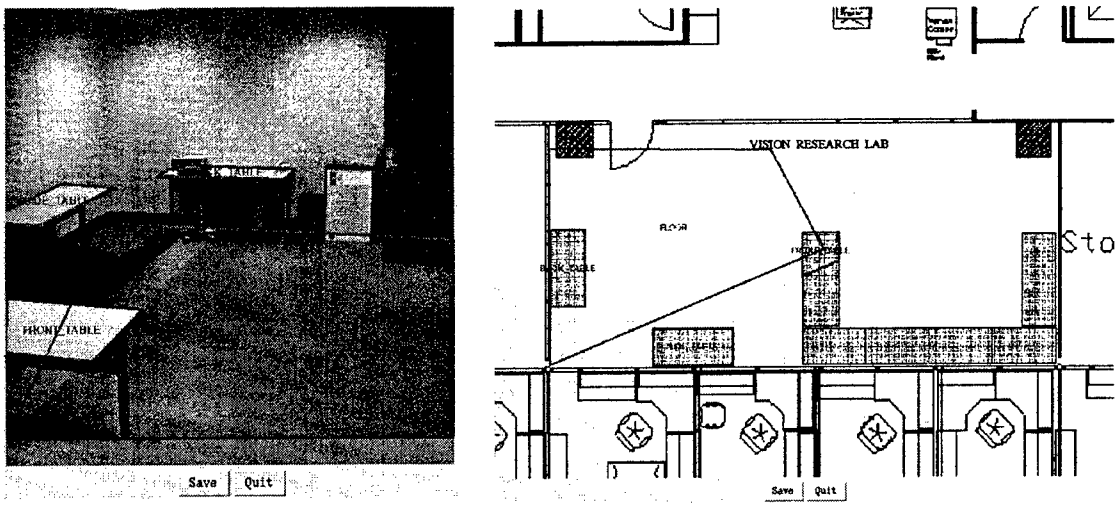


Figure 3: Establishing the image to map coordinate transformation

In a manner similar to the occlusion situation described above in section 3.2, the deposit event also gives rise to ambiguity as to which object is the depositor, and which is the depositee. For example, it may have been that the object which entered at frame F1 of Figure 2 *stopped* at frame F4 and deposited a *moving* object, and it is the deposited object which then proceeded to exit the scene at F6. Again, the AVS system relies on a moving vs. stationary distinction to resolve the ambiguity, and insists that the depositee remain stationary after a deposit event. The AVS system requires both the depositor and the depositee tracks to extend for n frames past the point at which the tracks separate (e.g., past frame F5 in Figure 2), and that the deposited object remain stationary; otherwise no deposit event is generated.

Also detected (but not illustrated in Figure 2), are REST events (when a moving object comes to a stop), and MOVE events (when a RESTing object begins to move again). Finally, one further event that is detected is the LIGHTSOUT event, which occurs whenever a large change occurs over the entire image. The motion graph need not be consulted to detect this event.

3.4 Image to World Mapping

In order to locate objects seen in the image with respect to a map, it is necessary to establish a mapping between image and map coordinates. This mapping is established in the AVS system by having a user draw quadrilaterals on the horizontal surfaces visible in an image, and the corresponding quadrilaterals on a map, as shown in Figure 3. A warp transformation from image to map coordinates is constructed using the quadrilateral coordinates.

Once the transformations are established, the system can estimate the location of an object (as in Flinchbaugh and Bannon [1994]) by assuming that all objects rest on a horizontal surface. When an object is detected in the scene, the midpoint of the lowest side of the bounding box is used as the image point to project into the map window using the quadrilateral warp transformation [Wolberg, 1990].

4 Applications

The AVS core algorithms described in section 3 have been used as the basis for several video surveillance applications. Section 4 describes three applications that we have implemented: situational awareness, best-view selection for activity logging, and environment learning.

4.1 Situational Awareness

The goal of the situational awareness application is to produce a real-time map-based display of the locations of people, objects and events in a monitored region, and to allow a user to specify alarm conditions interactively. Alarm conditions may be based on the locations of people and objects in the scene, the types of objects in the scene, the events in which the people and objects are in-

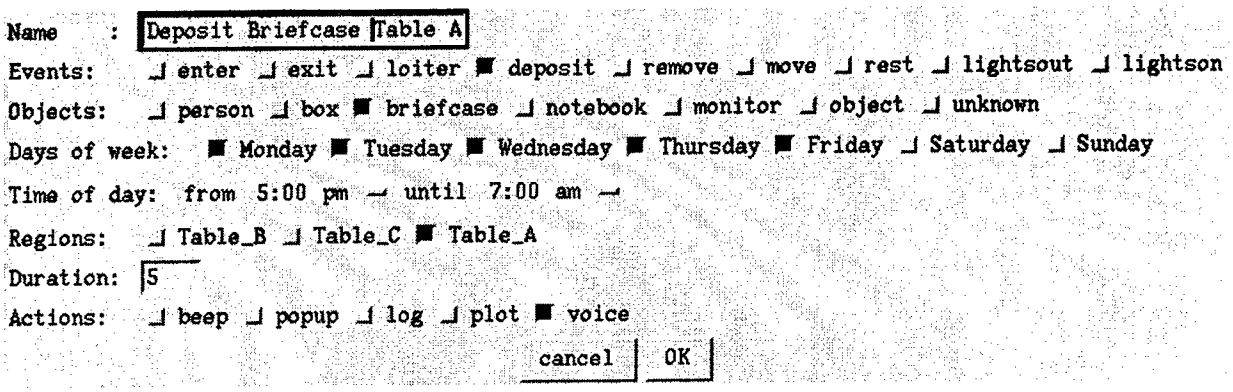


Figure 5: User interface for specifying a monitor in AVS

volved, and the times at which the events occur. Furthermore, the user can specify the action to take when an alarm is triggered, e.g., to generate an audio alarm or write a log file. For example, the user should be able to specify that an audio alarm should be triggered if a person deposits a briefcase on a given table between 5:00pm and 7:00 am on a weeknight.

The architecture of the AVS situational awareness system is depicted in Figure 4. The system consists of one or more smart cameras communicating with a Video Surveillance Shell (VSS). Each camera has associated with it an independent AVS core engine that performs the processing described in section 3. That is, the engine finds and tracks moving objects in the scene, maps their image locations to world coordinates, and recognizes events involving the objects. Each core engine emits a stream of location and event reports to the VSS, which filters the incoming event streams for user-specified alarm conditions and takes the appropriate actions.

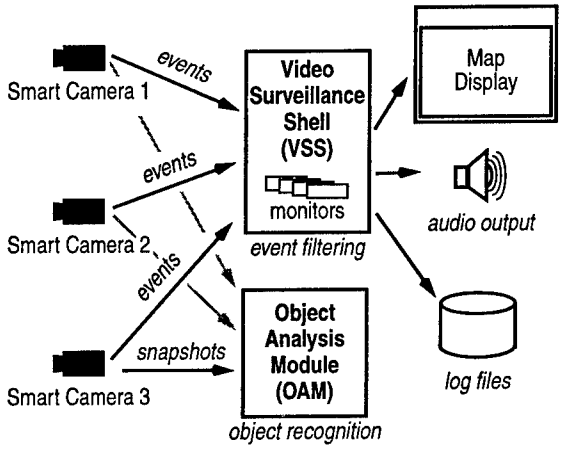


Figure 4: The situational awareness system

In order to determine the identities of objects (e.g., briefcase, notebook), the situational awareness system communicates with one or more object analysis modules (OAMs). The core engines capture snapshots of interesting objects in the scenes, and forward the snapshots to the OAM, along with the IDs of the tracks containing the objects. The OAM then processes the snapshot in order to determine the type of object. The OAM processing and the AVS core engine computations are asynchronous, so the core engine may have processed several more frames by time the OAM completes its analysis. Once the analysis is complete, the OAM sends the results (an object type label) and the track ID back to the core engine. The core engine uses the track ID to associate the label with the correct object in the current frame (assuming the object has remained in the scene and been successfully tracked).

The VSS provides a map display of the monitored area, with the locations of the objects in the scene reported as icons on the map. The VSS also allows the user to specify alarm regions and conditions. Alarm regions are specified by drawing them on the map using a mouse, and naming them as desired. The user can then specify the conditions and actions for alarms by creating one or more *monitors*. Figure 5 depicts the monitor creation dialog box. The user names the monitor and uses the mouse to select check boxes associated with the conditions that will trigger the monitor. The user selects the type of event, the type of object involved in the event, the day of week and time of day of the event, where the event occurs, and what to do when the alarm condition occurs. The monitor specified in Figure 5 specifies that a voice alarm

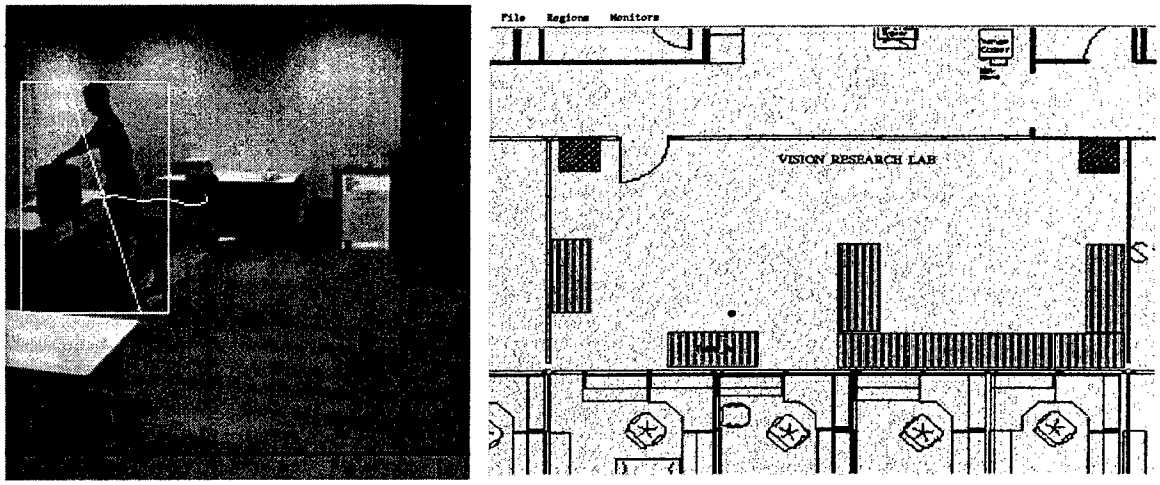


Figure 6: Tracking an object in the scene on the map

will be sounded when a briefcase is deposited on Table_A between 5:00pm and 7:00am on a weeknight. The voice alarms are customized to the event and object type, so that when this alarms is triggered, the system will announce "deposit box" via its audio output. Figure 6 shows a person about to trigger this alarm.

5 Best-View Selection for Activity Logging

In many video surveillance applications the goal of surveillance is not to detect events in real time and generate alarms, but rather to construct a log or audit trail of all of the activity that takes place in the camera's field of view. This log is examined by investigators after a security incident (e.g., a theft or terrorist attack), and is used to identify possible suspects or witnesses.

In order to gain experience with this type of application, we have used the tracking and event detection capabilities described in section 3 to construct a program that monitors and records the movements of humans in its field of view. For every person that it sees, it creates a log file that summarizes important information about the person, including a snapshot taken when the person was close to the camera and (if possible) facing it. The log files are made available to authorized users via the World-Wide Web.

5.1 Architecture

The application makes use of the AVS core algorithms to detect and track people. Upon detection of a track corresponding to a person in the input, the tracker associates a data record with the track. The data record contains a summary of information about the person, including a snapshot extracted from the current video image. As the person is tracked through the scene, the tracker examines each image of that person that it receives. If the new image is a better view of the person than the previously saved snapshot, the snapshot is replaced with the new view. When the person leaves the scene, the data record is saved to a file.

Each log entry file records the time when the person entered the scene and a list of coordinate pairs showing their position in each video frame. Each log entry file also contains the snapshot that was stored in the track record for the person when they exited the scene. Because of the way snapshots are maintained, the final snapshot is the best view of the person that the system had during tracking. Finally, the log entry file contains a pointer to the reference image that was in effect when the snapshot was taken. This information forms an extremely concise description of the person's movements and appearance while they were in the scene.

Selecting the best view: The system uses simple heuristics to decide when the current view of a per-

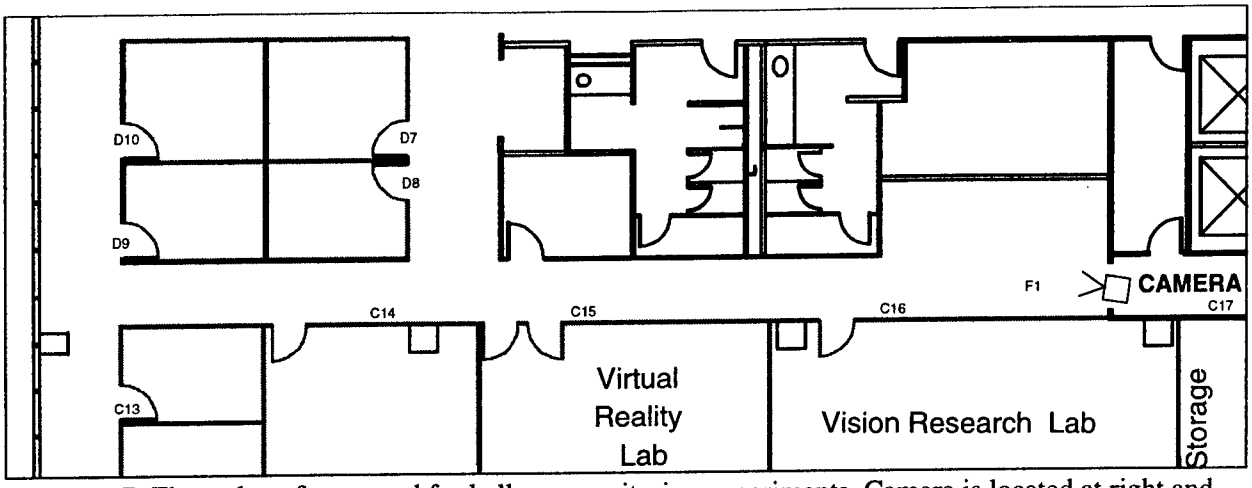


Figure 7: Floor plan of area used for hallway monitoring experiments. Camera is located at right and monitors the hallway and printer alcove.

son is better than the previously saved view. First, the new view is considered better if the subject is moving toward the camera in the current frame, and was moving away in the previously saved view. This causes the system to favor views in which the subject's face is visible. If this rule does not apply, the new view is considered better if the subject appears to be larger (subtends a larger visual angle). This causes the system to prefer views in which the subject is close to the camera. Other possible view selection heuristics are discussed in Kelly et al. [1995].

Handling background change: The test environment experiences significant lighting variation during the day due to window lighting, opening and closing doors etcetera. In addition, during the day people frequently deposit, remove, or reposition objects in the scene. This creates permanent regions of difference between the scene and the reference image. Without some mechanism for updating the reference image, the system would continue to track these difference regions as objects. Therefore, the tracker was instructed to discard the current tracks and grab a new reference image whenever it determined that all objects in the scene were stationary, and that no object had moved for several seconds.

User Interface

Log files are saved in a directory tree associated with the camera that produced the data. Along with the log files, the monitoring application creates HTML documents that allow a web browser to navigate the directory tree and access the log entries. Log entries are displayed by a Java applet that displays the best snapshot of the person in the context of the reference image, and overlays the person's path through the scene on the image. The applet runs as an independent thread that checks periodically to see if any new log entries have been created. Thus if the user is browsing the entries for the current day, new entries become available to the browser as soon as they occur.

5.2 Experiments

The system described above was tested in a hallway of our laboratory. Figure 7 shows the hallway floor plan. The camera is mounted in the hallway ceiling and looks west toward a window-lit corridor that runs around the perimeter of the building. The hallway experiences heavy traffic, because it contains a laser printer, a copier, and the office water cooler. The hallway passes under the camera and continues to the east out of the field of view.

The system was allowed to run for a total of 118 hours over a period of a week. Most laboratory personnel were unaware that a test was in progress, so the system was exposed to normal daily activity. During the test the system recorded a total of 965 log entries. Figure 8 shows the browser display for a typical log entry. In this sequence the subject entered the scene from the cross corridor at rear and came down the hallway on his way to the copier, out of view at lower right. His path is shown as a line on the floor, which appears red when viewed with a color browser.

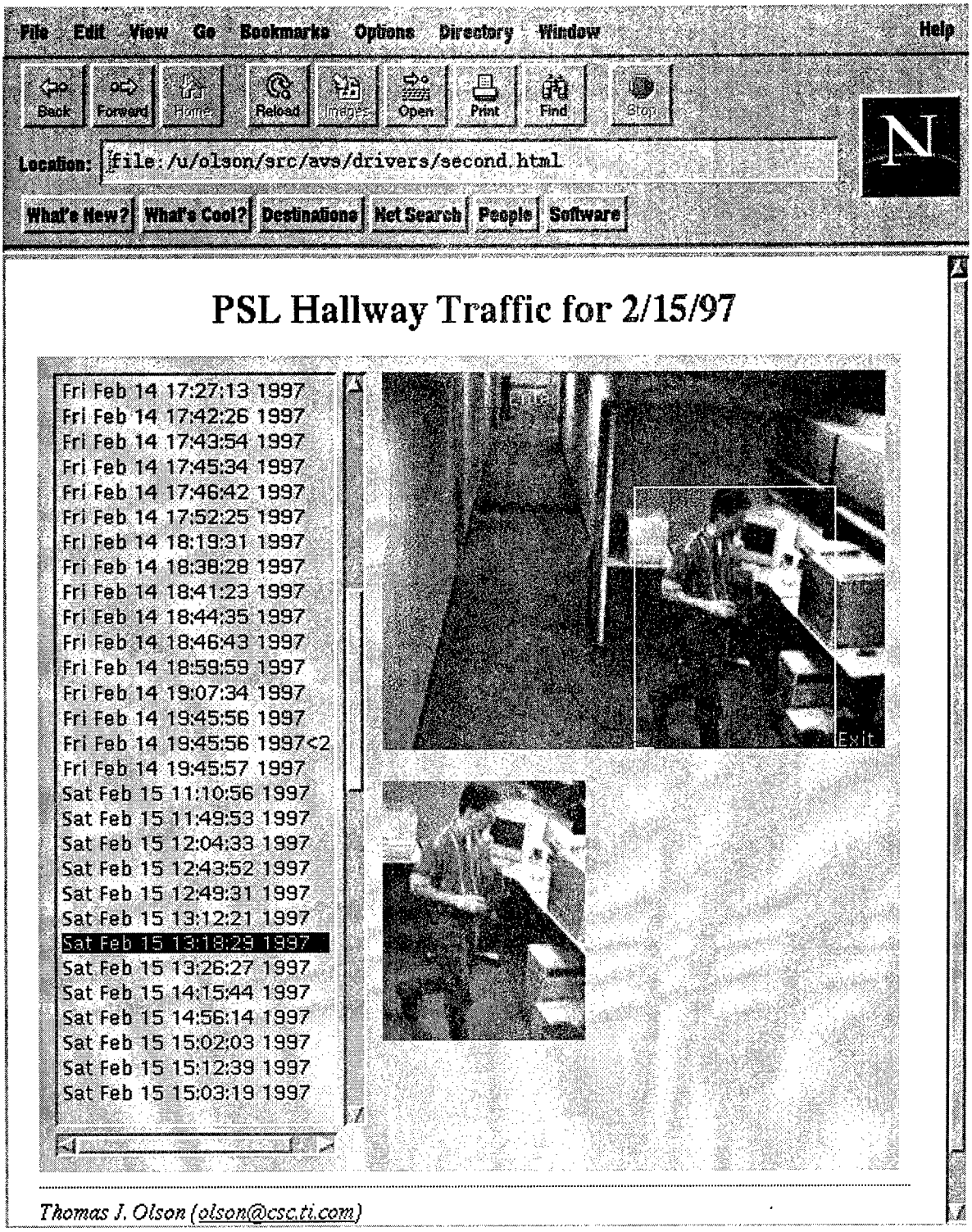


Figure 8: Log entry browser interface. The line drawn on the floor in the upper image shows the subject's path from entry to exit. The list entry selected at left is the time at which the image was taken.

Figure 9 demonstrates the effect of the system's preference for frontal views. In this sequence the subject entered at the bottom of the scene and walked away from the camera. He turned around and took a few steps back toward the camera, then turned away again and continued down the hallway, eventually exiting via the first door on the left. Although the subject's back was toward the camera

most of the time, the view preference heuristics selected a view taken while he was facing the camera.

Performance Evaluation

In order to assess the performance of the monitoring application, all of the log entries for the experiment period were examined and scored by one of the authors. Entries were classified as follows:

Face/Non-face: Entries containing a view of a subject's head were classified as FACES if the subject's face (specifically, subject's nose) was visible, otherwise they were classified as NONFACES.

False Alarm: Images which contained no human and appeared to be caused by noise were classified as FALSE ALARMS.

Bad Path: Entries in which the floor trace is clearly corrupt in some way were classified as BAD PATHs.

Bad Choice: In some cases it is clear from the floor trace that the system made a poor choice of which image of a person to save in the log entry. These entries were classified as BAD CHOICE.

False Negative: In some cases it is clear that the system failed to take a usable picture of a person who was in the scene. These were classified as FALSE NEGATIVES. About half of the false negatives occurred when the system selected a view in which the subject's head is not visible, typically because they were in the act of passing through a doorway. The others occurred when the system became confused by occlusion, and incorrectly grouped two people into a single log entry. Note that we do not have ground truth for the observation period, so there may have been other detection failures that were not detected. However, monitoring by the authors during the daytime revealed no failures of this type. We believe that the FALSE NEGATIVE count is a good estimate of the number of detection failures.

Table 1 shows the classification counts for the test period. Assuming that the false negative count is

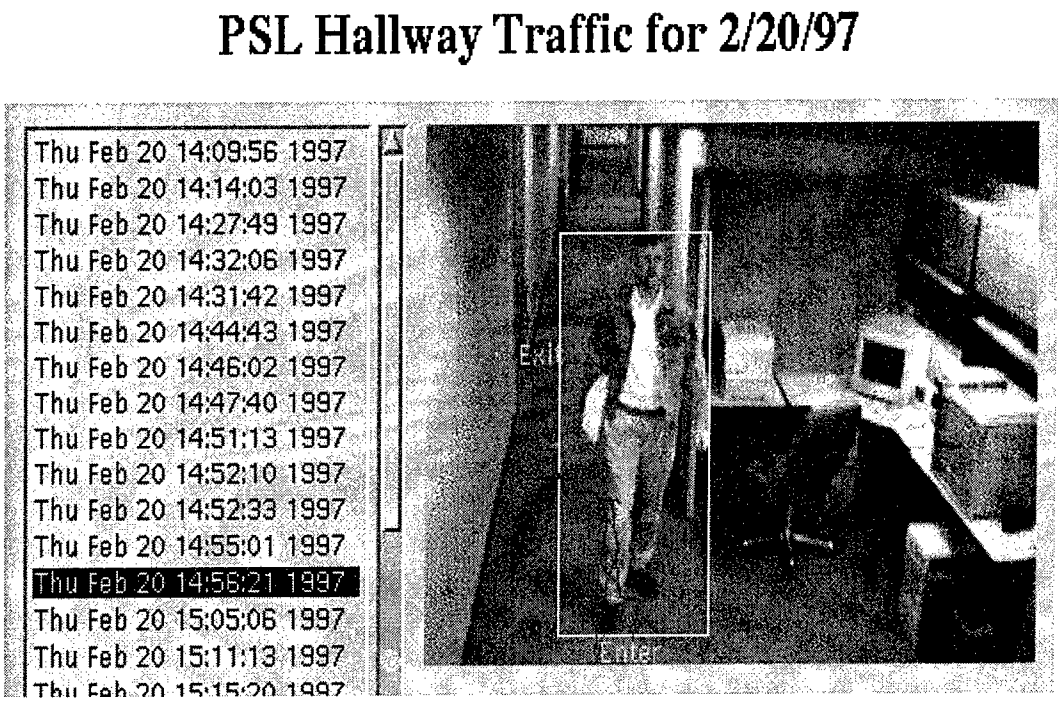


Figure 9: Log entry showing the effect of the view selection heuristic preference for frontal views. The subject was walking away from the camera for most of this sequence, but the system was able to capture a view while he was facing the camera.

Table 1: Long-term monitoring system performance

log entry type	Number of entries
FACE	493
NONFACE	380
FALSE ALARM	62
FALSE NEGATIVE	44
BAD PATH	112
BAD CHOICE	29
TOTAL ENTRIES	965

valid, the system achieved a detection rate of 95.2% with a false alarm rate of 6.4%. The recorded path of the subject was correct (or at least plausible) in 88.4% of entries, and the system made conspicuously bad choices of what image to save in only 3% of entries.

Of the valid images of humans, 56.6% showed the subject's face, vs. 43.4% that did not. Note that in most cases where the image does not show the face, the subject entered the scene from below the camera and walked away from it, so there was never an opportunity for a frontal view. Earlier experiments without the frontal view heuristic captured FACE and NONFACE images with roughly equal frequency, so the it is clear that the heuristic helps.

At the end of the experiment, the camera directory occupied 34.5 megabytes, or about seven megabytes per day of monitoring. Almost all of the storage consists of image files, so presumably compression with an image-specific algorithm would produce substantial savings. Use of an MPEG-like algorithm on the reference images would be extremely effective, since the reference images are all very nearly identical, and lossless compression would not be necessary.

6 Learning Environment Structure

The AVS tracking and event recognition software uses corresponding rectangles in image and world coordinates to compute an approximate image-to-world mapping. These rectangles are created by a human when the camera system is set up. In many situations it would be preferable to eliminate even this minimal calibration step, in order to reduce setup cost to a minimum.

We have developed a system that learns the image-to-world mapping by watching humans move around in the scene. Changes in the apparent size and position of humans in the image provide information about the existence and depth of world surfaces. Appearance and disappearance of humans provides information about occlusion boundaries and locations where humans can enter or exit the scene.

6.1 Method

The computation assumes weak perspective projection, i.e. that objects in the scene are first projected orthographically to a plane passing through a reference point on the object and parallel to the image plane, and then projected to the image plane using true perspective. It is also assumed that humans are usually in contact with a world surface that supports them, that the camera is in an upright position (has roll angle zero), and that the internal calibration parameters of the camera are known.

More precisely, assume front projection with the camera focal point at the origin and looking down the Z axis of a left-handed coordinate system. Suppose the camera observes a person in the world with head at world point $H = (X_H, Y_H, Z_H)$ and feet at world point F. Let F be the reference point for weak perspective projection. Then the apparent height of the person in the image is given by

$$y_H - y_F = \frac{f}{Z_F}(Y_H - Y_F) = \frac{f}{Z_F}|H - F|\cos\theta$$

where θ is the camera tilt angle relative to the local vertical direction. Solving for depth gives

$$Z_f = f\cos\theta\left(\frac{|H - F|}{y_h - y_f}\right)$$

The person's height $|H - F|$ has a known probability distribution, and the tilt angle term $\cos\theta$ can be

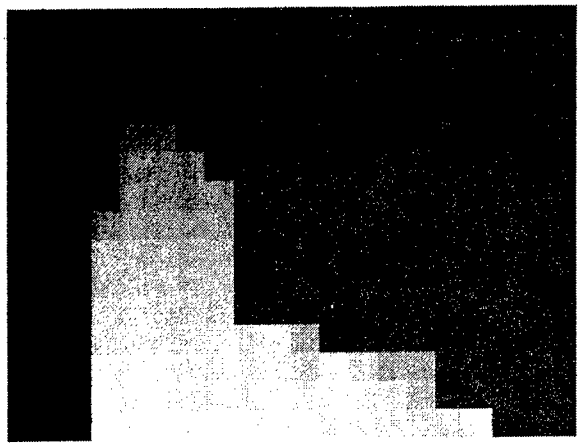


Figure 10: Apparent height data collected in the experiment. Cell intensity is the median of the image heights of observed humans when their feet were imaged in the cell. Dark grey regions contain no data.

estimated from the appearance of the person, or simply ignored for the shallow tilt angles typical of security camera installations. Given enough observations, the equation can be used to estimate the distance from the camera to points in the world where people commonly walk.

The idea of recovering structure from observed sizes of humans is conceptually related to shape-from-texture work in which the texture is made up of discrete elements that are uniform in size and shape [Aloimonos and Swain, 1988, Blostein and Ahuja, 1989]. In this case the texels (people) do not lie in the imaged surface, and their size in the world is known. This makes depth recovery substantially easier than it is in general shape-from-texture work.

6.2 Mapping the Environment

The equation derived above has been used in a program that learns the structure of its environment by watching humans move around in it. The program makes use of the AVS core algorithms to detect and track people. Over time, it builds up an image in which pixel value represents depth to the nearest world surface in the corresponding direction.

The camera image is partitioned into a grid of 16x16-pixel squares, each of which is associated with a histogram. Whenever the program detects a person in the scene, it locates the histogram associated with the place where they are standing, i.e., the one associated with the square containing the bottom center of the motion region for the person. The apparent height of the person is recorded in that histogram. Over time, the histogram for each location in the image builds up a sample distribution for the apparent (image) height of humans at that location. This can be used with the equation derived previously to estimate the depth at that point.

The program was allowed to operate for twenty-four hours during a typical working day. Input was provided by the hallway camera used in section 5. Figure 10 shows the raw output of the program. In the figure pixel intensity corresponds to the median observed height for the corresponding location. Dark grey pixels are those for which no observations were recorded. The program was instructed to discard observations in which the motion region for the person touched the upper or lower image border, since the apparent height is invalid in that condition. For this reason, there are no counts for the end of the hallway.

The height data of Figure 10 were converted to depths using the equation derived above. Vertical pixel pitch was taken from the camera technical manual, and the nominal lens focal length was used to approximate the true focal length. Histogram cells for which fewer than ten total observations were recorded were discarded.

Figure 11 shows the final depth map superimposed on the image. The range estimates cover image regions corresponding to the floor, and vary smoothly over most of the image. Anomalously large values occur in several locations at right center below the small printer and workstation. These errors occur because the office chair is frequently moved around in this region, and the system sometimes mistakes it for a person. Since it is significantly smaller than a real person, the system interprets it as evidence that the floor supporting it is further away than it actually is. A similar problem produces the anomalously high value of 8.9 meters at left center, at the base of the doorway. It frequently happens that as a person exits the hall via the doorway, their head goes out of sight while their body and feet are still visible. The system records the height of the visible portion of the person in the cell at the base of the doorway. Since this

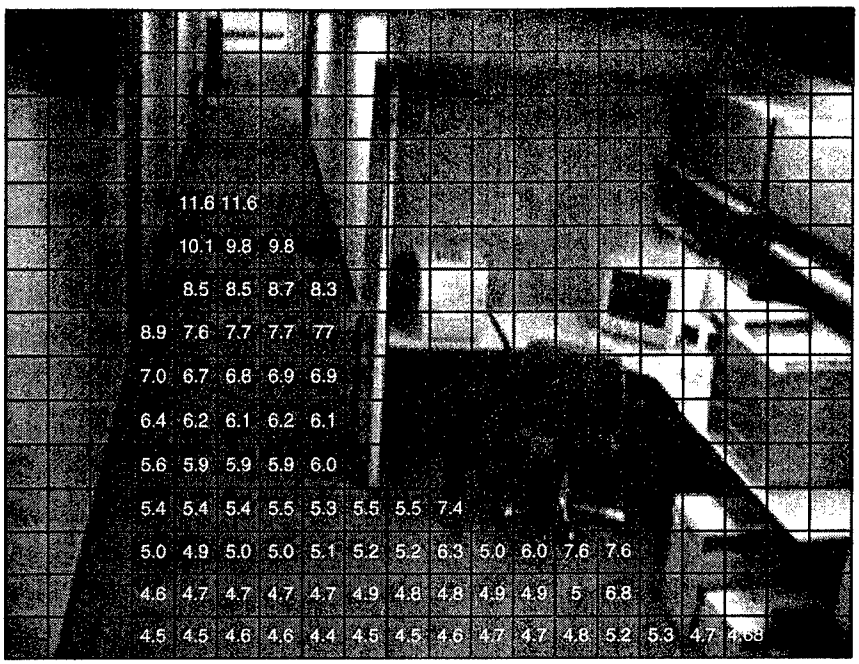


Figure 11: Depth map recovered from the height data of figure 10. Depths are in meters.

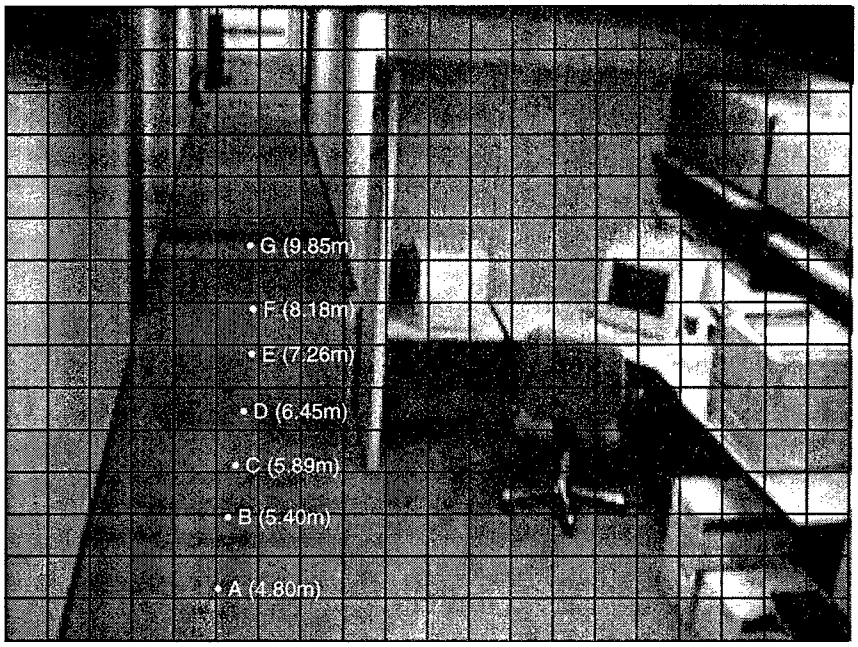


Figure 12: Ground truth range values for comparison with figure 11.

value is smaller than the true height of the person, that cell appears to be further away than it really is.

In order to assess the accuracy of the recovered depth map, we measured the distance from the camera to seven points on the floor. The seven points and their distances from the camera are shown superimposed on the image in figure 12. Table 2 shows the estimated and actual ranges to the test points, as well as the error in meters. The average absolute error for the seven test points is 26cm, which is less than 5% of the average distance.

Table 2: Estimated vs. Actual Range (meters) to ground truth points

point	estimate (meters)	actual (meters)	error (meters)
A	4.70	4.80	-0.10
B	5.00	5.40	-0.40
C	5.90	5.89	0.01
D	6.10	6.45	-0.35
E	6.80	7.26	-0.46
F	7.70	8.18	-0.48
G	9.80	9.85	-0.05

7 Conclusion

The goal of our research is to develop algorithms and systems that can be used to describe a video sequence in terms of moving objects and events. These algorithms will enable a generation of smart cameras that deliver information about scenes rather than raw images. We have created a set of core algorithms comprising the Autonomous Video Surveillance (AVS) system, including routines for moving object detection, tracking, and abstract event recognition. The AVS system has been used to create several surveillance applications, including a video surveillance shell, a program that creates concise logs of activity in the field of view, and a program that learns scene structure by watching humans moving around in the environment.

Our future work on AVS will address weaknesses in the current system, and will add new capabilities that support more complex applications. Work is planned in three main areas:

Robust Change Detection and Tracking: Experiments have shown that errors in the moving object detection computation are the most common cause of errors in our applications. This is particularly a problem in outdoor environments. We plan to develop new change detection algorithms based on dynamic background models that capture the way the background changes over time. We will also exploit contextual information to predict the expected size and appearance of moving objects in the scene.

Improved Event Recognition: We will extend our motion-graph-based event recognition algorithms to a broader range of events, and will develop methods of specifying and recognizing compound events and event sequences.

Applications: We will extend the existing video surveillance shell to make use of authentication sensors, and to distinguish between authorized and unauthorized individuals. We will continue to use AVS technology to develop applications that address military and other government video surveillance needs.

References

[Aloimonos and Swain, 1988] Y. Aloimonos and M. Swain. Shape from texture. In *Proc Ninth International Joint Conf. on Artificial Intelligence*, Los Angeles, pp. 926-931, 1988.

[Blostein and Ahuja, 1989] D. Blostein and N. Ahuja. Shape from texture: Integrating Texture-Element Extraction and Surface Estimation. *IEEE. Trans. Pattern Analysis and Machine Intelligence*, v. 11 no. 12, pp. 1233-1251, Dec. 1989.

[Bobick and Davis, 1996] A. Bobick and J. Davis. Real-time recognition of activity using temporal templates. In *Proc. Third IEEE Workshop on Applications of Computer Vision*, pp. 39-42, 1996.

[Burt et al., 1988] P. Burt, J. Bergen, R. Hingorani, R. Kolczynski, W. Lee, A. Leung, J. Lubin, and H. Shvayster. Object tracking with a moving camera. In *Proc. IEEE Workshop on Visual Motion*. pp. 2-12, Irvine, March 1988.

[Camus et al, 1993] T. Camus, J. Monsarrat and T. Dean. Planning and Selective Perception for Target Retrieval. In *Proc. DARPA Image Understanding Workshop*, pp. 593-597, April 1993.

[Courtney, 1997] J. D. Courtney. Automatic video indexing via object motion analysis. *Pattern Recognition*, **30**(4), April 1997.

[Flinchbaugh, 1997] B. Flinchbaugh. Robust Video Motion Detection and Event Recognition. in *1997 Proc. DARPA Image Understanding Workshop* (these proceedings).

[Flinchbaugh and Bannon, 1994] B. Flinchbaugh and T. Bannon. Autonomous scene monitoring system. In *Proc. 10th Joint Government-Industry Security Technology Symposium, American Defense Preparedness Association*, June 1994.

[Gavrila and Davis, 1996] D. Gavrila and L. Davis. Tracking of humans in action: A 3-D model-based approach. In *1996 Proc. DARPA Image Understanding Workshop*, pp. 737-746, 1996.

[Intille and Bobick, 1995] S. Intille and A. Bobick. Closed-world tracking. In *Proc. Fifth International Conference on Computer Vision*, pp. 672-678, 1995.

[Jain et al., 1979] R. Jain, W. Martin, and J. Aggarwal. Segmentation through the detection of changes due to motion. *Computer Graphics and Image Processing*, **11**, pp. 13-34, 1979.

[Kahn et al., 1996] R. Kahn, M. Swain, P. Prokopowicz, and J. Firby. Gesture recognition using the Perseus architecture. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 734-741, 1996.

[Kakadiaris and Metaxas, 1996] I. Kakadiaris and D. Metaxas. Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 81-87, San Francisco, 1996.

[Karmann and von Brandt, 1990] K. P. Karmann and A. von Brandt. Moving object recognition using an adaptive background memory. In *Time Varying Image Processing and Moving Object Recognition*, V. Cappellini (ed.). **2**, pp. 1060-1066, Elsevier, Amsterdam, 1990.

[Kelly et al., 1995] P. Kelly, A. Katkere, D. Kuramura, S. Moezzi, S. Chatterjee and R. Jain. An Architecture for Multiple Perspective Interactive Video. Visual Computing Laboratory Technical Report VCL-95-103, UCSD, March 1995.

[Koller et al., 1994] D. Koller, J. Weber, J. Malik. Robust multiple car tracking with occlusion reasoning. In *Proc. Third European Conference on Computer Vision*, pp. 186-196, LNCS 800, Springer-Verlag, May 2-6, 1994.

[Makarov, 1996a] A. Makarov. Comparison of background extraction based intrusion detection algorithms. In *Proc. Int'l Conference on Image Processing*, Lausanne, Sept. 1996.

[Makarov et al., 1996b] A. Makarov, J. M. Vesin, and F. Reymond. Intrusion detection robust to slow and abrupt lighting changes. In *Proc. SPIE: Real-Time Imaging*, **2661**, pp. 44-54, 1996.

[Nelson, 1991] R. C. Nelson. Qualitative detection of motion by a moving observer. *International Journal of Computer Vision*, **7**(1), pp. 33-46, November 1991.

[Pentland, 1996] A. Pentland. Machine understanding of human action. In *Proc. DARPA Image Understanding Workshop*, pp. 757-764, 1996.

[Rimey and Brown, 1993] R. Rimey and C. Brown. Studying Control of Selective Perception with T-world and TEA. In *Proceedings of the DARPA Image Understanding Workshop*, Washington D.C., pp. 575-580, April 1993.

[Ringler and Hoover, 1995] C. Ringler and C. Hoover. Evaluation of commercially available VMDs. Sandia National Laboratories Technical Report SAND94-2875, June 1995.

[Rowley et al., 1996] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *Proc. DARPA Image Understanding Workshop*, pp. 725-735, 1996.

[Starner and Pentland, 1995] T. Starner and A. Pentland. Visual recognition of American Sign Language using hidden Markov models. In *Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, 1995.

[Sung and Poggio, 1994] K. K. Sung and T. Poggio. Example-based learning for view-based human face detection. In *Proc. DARPA Image Understanding Workshop*, pp. 843-850, 1994.

[Swain and Stricker, 1991] M. J. Swain and M. Stricker, editors. Promising directions in active vision. University of Chicago Technical Report CS 91-27, November 1991.

[Wolberg, 1990] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA, 1990.

[Yalamanchili et al., 1982] S. Yalamanchili, W. Martin, and J. Aggarwal. Extraction of moving object descriptions via differencing. *Computer Graphics and Image Processing*, **18**, pp. 188-201, 1982.

Sensory Attention: Computational Sensor Paradigm for Low–Latency Adaptive Vision

Vladimir Brajovic and Takeo Kanade
School of Computer Science, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh PA 15213
brajovic@cs.cmu.edu, tk@cs.cmu.edu
http://www.cs.cmu.edu/afs/cs/usr/brajovic/www/lab/vlsi.html

Abstract[1]

The need for robust self–contained and low-latency vision systems is growing: high speed visual servoing and vision–based human computer interface. Conventional vision systems can hardly meet this need because 1) the latency is incurred in a data transfer and computational bottlenecks, and 2) there is no top–down feedback to adapt sensor performance for improved robustness. In this paper we present a tracking computational sensor — a VLSI implementation of a sensory attention. The tracking sensor focuses attention on a salient feature in its receptive field and maintains this attention in the world coordinates. Using both low –lateny massive parallel processing and top–down sensory adaptation, the sensor reliably tracks features of interest while it suppresses other irrelevant features that may interfere with the task at hand.

1. Introduction

The computational sensor paradigm [Kanade and Bajcsy, 1993] has the potential to greatly reduce latency and provide top–down sensory adaptation to vision systems. By integrating sensing and processing on a VLSI chip, both transfer and computational bottlenecks can be alleviated; on–chip routing provides high throughput transfer, while an on-chip processor could implement massively–parallel fine–grain computation, thus providing high processing capacity which readily scales up with the image size. In addition, the tight coupling between processor and sensor allows for efficient top–down feedback that can control and adjust sensor for further acquisition based on the preliminary results of the processing. Our recent work has been concerned with efficient implementation of global operations over a large group of image data using the computational sensor paradigm [Brajovic and Kanade, 1994]. We have formulated two mechanisms for implementing global operations in computational sensors: (1) *intensity–to–time processing paradigm* [Brajovic and Kanade, 1996], and (2) *sensory attention* presented in this paper.

2. Approach

The sensory attention is based on the premise that salient features within the retinal image represent important global features of the entire image. By selecting a small region of interest around the salient feature for subsequent processing, the sensory attention eliminates extraneous information and allows the processor to handle small amounts of data at a time. We have implemented sensory attention by fabricating and testing *tracking computational sensor.* The tracking computational sensor optically receives a saliency map and continuously selects and tracks the peaks in it. The location and intensity of the selected peaks is reported on few output pins with low latency. These quantities are also used internally in a top–down fashion to aid tracking of the attended location. The chip is a 28 x 28 array of 60μ x 60μ cells, and is fabricated on a 2.2mm x 2.2mm die.

The *sensory attention* follows the model of *visual attention* in brains. This analogy is attractive for

1. This research has been sponsored by Office of Naval research (ONR) under Contract N00014–95–1–0591. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ONR or the U.S. Government.

two reasons. First, the main argument that has been used to explain the need for selective visual attention in brains is that there exist some kind of processing and communication limitation in the visual system. So it does in machines. Attention "funnels" only relevant information and protects the limited communication and processing resources from the information overload. Second, it has been shown that the visual attention improves performance, and is needed for maintaining coherent behavior while interacting with the environment (i.e., attention–for–action) [Allport, 1989]. Unlike eye movement (i.e., *overt* shifts), the attention shifts (i.e., *covert* shifts) do not require any motor action, but occur internally on a fixed retinal image. For this reason, attention shifts are faster and play an important role in low–latency vision systems.

It is interesting to note that foveating computational sensors [Kanade and Bajcsy, 1993] try to emulate this kind of data compression. For example, Van der Spiegel's log–polar sensor samples images within fovea with high acuity, while maintaining sparse representation at the periphery. This sensor simulates overt shifts, since it requires motor action for foveating. Kosonocky's foveating sensor allows programmable fovea within the retinal image; therefore, it eliminates the need for mechanical action and simulates covert shifts. Another related solution is random access to the image data. For example, Laval's MAR sensor attends to, and reads only, a small local portion of the retinal image, the part that is necessary for the local convolution performed in the global off–chip processor. However, these computational sensors act as special cameras and the mechanism which guides the location of the attention is missing.

To apply attention selection in machines, several issues must be solved: (1) the problem of selecting an "interesting" location, (2) the problem of shifting to another location, and (3) the problem of transferring local data for further processing. In a very influential paper [Koch and Ullman, 1987], Koch and Ullman address these issues. The selection process utilizes a *saliency map* that encodes conspicuousness or the level of interest throughout the retinal image. The saliency map can be derived from image features, including: intensity, color, spatial and temporal derivatives, motion, and orientation. For selecting a location of the attention within the saliency map, *winner–take–all* (WTA) mechanism has been suggested. The WTA is not responsible for information processing; rather it determines only which area of the retinal image should be relayed to the global processor for further inspection.

The problem of shifting to another location is somewhat more challenging. It is observed in humans that interesting visual stimulation initially (i.e., during the first 100ms) captures the attention; later (i.e., after 300ms) it has inhibitory effects which can last up to 1.5 seconds [Milanese, 1993]. The inhibitory effect prevents the subject from returning to previously visited locations. The inhibition is "stored" in environmental coordinates rather than in image coordinates; therefore, reliable operation is maintained even in the presence of ocular or object movement. The attention shifts can be initiated on a voluntary basis by telling the observer the location of a target, or they can be automatic caused by the onset of a visual stimulus. For shifting to another location, Koch and Ullman's model allows the saliency of the currently attended location to decay, even if the visual stimuli creating the saliency remain present. This will release the WTA mechanism and allow it to converge to another location. Either a *local* or *central* inhibition mechanism for initiating decay is possible. The local mechanism causes the saliency to decay some time after the WTA has converged to a particular location. In the central mechanism, once the attended portion of the retinal image is relayed to the central processor, a signal, which inhibits the conspicuousness of the currently attended location, is sent back. The local inhibition mechanism mimics the automatic attention shift, while the central mechanism can initiate voluntary attention shifts.

Recently, Morris et al. [Morris and DeWeerth, 1996] reported an analog VLSI circuit implementation of covert attention shifts as suggested by the Koch and Ullman model. A one–dimensional 19 cell circuit implements: 1)saliency map normalization, 2)WTA location selection with preference for spatial proximity shifts, 3)inhibition of return control and 4) position detection for producing the location of attention as the output. Depending on the biasing condition, the circuit is able to roam between the peaks in the stationary saliency map.

In *the attention–for–action* model, Allport sug-

gested that attention goes beyond protecting the limited processing resources during complex object recognition: *attention is needed to ensure behavioral coherence* [Allport, 1989]. Since visual perception is the means for allowing a subject to interact with the environment (e.g., manipulate, avoid, etc.), it must produce actions consistent with the subject's goals. Selective processing is necessary in order to isolate the information that defines parameters for the appropriate action. For example, to catch a moving object, among many other moving and stationary objects, the information specific only to that object determines the action. Information about other objects in the visual field must be kept from interfering with the goal of catching the target object, even though other objects may influence how the target object is caught. In other words, attention aids the target goal by masking the irrelevant information's interference, but allows the action to be modified or diverted if new, important events occur.

The attention–for–action model is in close agreement with our goal of producing reliable *low–latency* computational sensors which provide *useful* information for the *coherent interaction with the environment*. It is not hard to imagine that if the attention is allowed to arbitrarily roam from one location to another, as suggested by Koch and Ullman's model and implemented in [Morris and DeWeerth, 1996], it may take a long time before the global processor encounters the *relevant* information for an appropriate action. We need more control over attention shifts, possibly by employing the *central inhibition* mechanism in combination with the *voluntary focus of attention* directed toward desired goals. For robust operation, such shifts must maintain the location of attention in the presence of ocular or object motion [Milanese, 1993].

3. Implementation

In the prototype implementation of the sensory attention proposed by this work, our concern is not how to compute the saliency map, but rather how to quickly and reliably locate and maintain an interesting location in the saliency map. We call this embodiment of the sensory attention *tracking computational sensor* because, when the saliency map is a natural image — the trivial saliency map — the features that attract attention are bright spots in the environment. The tracking computational sensor selects and tracks those spots while ignoring the background.

3.1. Location Selection

An image representing a saliency map is focused onto the array of photo detectors: photodiodes or photo transistors. The generated photo currents are fed to the winner–take–all (WTA) circuit which is responsible for the detection of the maximum point. The selected location is called a *feature*. Our design is based on a WTA circuit originally proposed in [Andreou et al. 1992] and [Lazzaro et al., 1988] shown is Figure 1. Currents $I_1 \ldots I_N$ are the input photo currents, while currents $J_1 \ldots J_N$ are the outputs of the WTA circuit. The cell receiving the largest photo current $I_k = max(I_1 \ldots I_N)$ responds with non–zero output current $J_k = I_c \neq 0$, while other cells respond with zero currents, i.e., $J_i = 0$, for $i \neq k$. The peak photo current establishes and holds the common voltage V_c. For small input currents, like those produced by light detection, the transistor operates in the subthreshold region. In that case, the voltage V_c is the logarithm of the winning input current: $V_c = V_o \log(I_1/I_o)$, where I_o is the process parameter and $V_o = kT/q\kappa$. Therefore, the intensity of the winner is accessed globally by monitoring the voltage on the common wire.

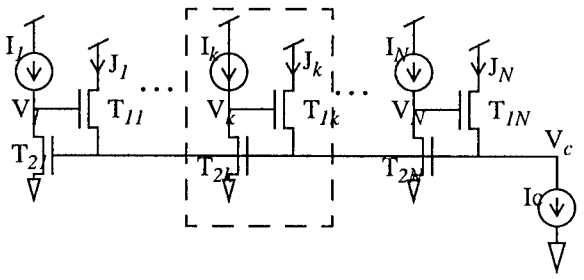


Figure 1: Schematic diagram of the winner–take–all circuit. Boxed area indicates one cell.

Since only the winning cell responds with non–zero current, the WTA effectively provides 1–of–N binary encoding of the winner's position. A digital on-chip decoder easily converts this code to any other binary code such as a natural binary or BCD code. In addition, there are efficient analog means for winner localization [DeWeerth, 1992]. In one example, the outputs from each WTA cell are con-

nected to nodes of a linear resistive network. The network behaves as a current divider splitting current I_c into two peripheral components, each proportional to the position of the current injection. By reading these peripheral components, the location of the winning cell is found. The WTA cells can be physically laid out in a two–dimensional array. Using the method of projections [Horn, 1986], the position of this current in two dimensions is found by solving two one–dimensional problems. Two copies of the output current are summed into the horizontal and vertical bus, respectively. The total current in these buses represents the desired projections onto the x and y axes. Then, two linear resistive networks are used at the periphery of the array to locate the winner in a x and y direction.

3.2. Location Shifts

The two dimensional WTA circuit locates the absolute maximum in the entire saliency map. In practical applications, there are often several strong features in the saliency map which are candidates for attracting the attention. For implementation of the attention–for–action model, we need to direct attention toward a feature that is useful for the task at hand. This corresponds to voluntary attention shifts, i.e., "telling" the sensor where to "look." Once the feature is selected, we need a mechanism that will track the feature and thus maintain the location of attention in the environmental coordinates even in the presence of ocular motion. Our implementation inhibits portions of the saliency map and restricts the activity of the WTA circuit within a programmable active region within the whole array of photo receptors. The active region is programmed by appropriate row and column addressing, and corresponds to the central inhibition control suggested by Koch and Ullman.

There are two modes of operation: (1) select mode, and (2) track mode. In the *select mode*, the active region is defined by the external addressing (Figure 2a). The active region can be of arbitrary size and location. The sensor selects the absolute maximum within this region. In the *tracking mode* the sensor itself dynamically defines a small (e.g., 3 x 3 in our implementation) active region centered around the most recent location of the attention (Figure 2b).

The *select* mode directs the attention towards a feature that is useful for the task at hand. For example, a user may want to specify an initial active region, aiding the sensor to attend to a relevant local peak in the saliency map. Then, the *tracking* mode is enabled for locking onto the selected feature. The ability of the sensor to define its own active region is an example of the top–down sensory adaptation presently missing from conventional vision systems.

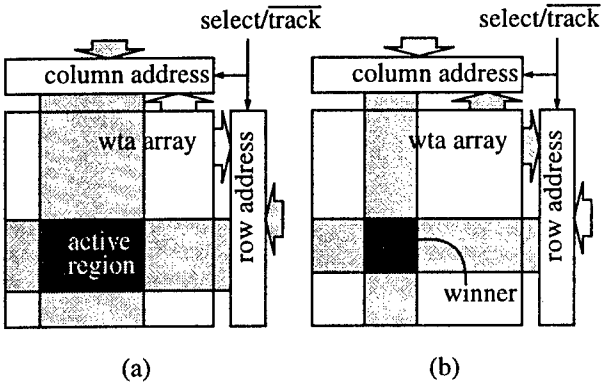


Figure 2: Modes of operation for the sensory attention computational sensor: (a) select mode, and (b) tracking mode.

The active region is programmed by inhibiting particular WTA cells under the external control. A circuit diagram of the WTA cell with inhibition is shown in Figure 3. The shunting path for the photo current is provided through the transistors T_5 and T_6. To maintain the cell active both $\overline{col}$ and $\overline{row}$ signals must be asserted (i.e. must be zero).

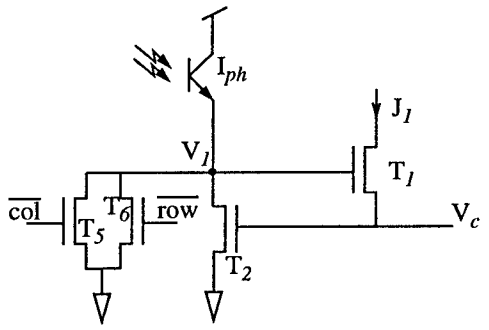


Figure 3: WTA cell with inhibition. (Shadowed area indicates components for cell inhibition.)

The control of active region is achieved from the periphery of the two–dimensional WTA array. The peripheral logic across three columns is shown in Figure 4. Similar logic is implemented for row addressing. In the select mode, the active column band is programmed by the content of the shift reg-

ister. There are no restrictions on the width or location of the band, as any bit pattern can be entered into the shift register.

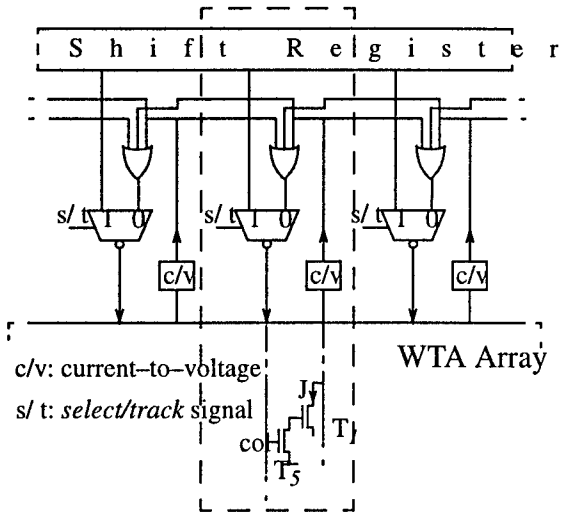


Figure 4: Peripheral logic for central control of the active region. The boxed area indicates one column. Similar logic is used for row addressing (not shown).

In the track mode, the active region is programmed by the WTA array and is dependent on the location of the feature being tracked. A particular column is enabled if the winning feature is on that column, or on one of the two immediate neighbors. In conjunction with the row inhibition (not shown), the tracking mode programs a 3 x 3 active region centered on the most recent feature. If that feature starts moving, one of the eight active neighbors will receive the winning feature and automatically update the position of the 3x3 active region. It is now clear that the salient feature is not necessarily the absolute maximum in the field–of–view, but rather it is a local peak in the retinal image. If for any reason the tracking mode starts on a location which is not a local peak, the 3x3 active region will "slide" along the intensity gradient until it locks onto a nearby peak.

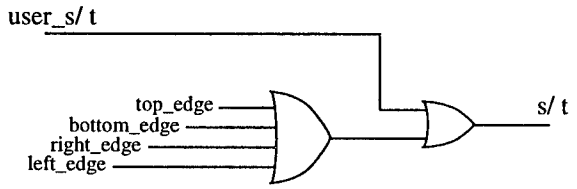


Figure 5: Logic for automatic switching between *select* and *track* modes.

With moving objects, the feature which is being tracked may reach the sensor's edge and fall out of the field of view. In order to ensure coherent transition in these situations, the logic shown in Figure 5 is implemented. The user may define the select mode by asserting signal *user_s/t*. However, when the user enables the tracking mode, the active region will be of size 3 by 3 as long as the tracked feature is not on one of the four edges of the array. When the feature reaches one of the four edges, the sensor automatically goes to a select mode. For a moment, the active region specified in the shift registers is enabled, and the absolute maximum is selected therein. If the newly selected feature is no longer on the edge, the sensor automatically goes back to the tracking mode, shrinks the active region to a 3 by 3 size, and continues feature tracking.

3.3. Transferring Local Data

Once the relevant conspicuous point has been localized in the saliency map, the local data from the attended vicinity must be transferred to the global processor for decision making. The local data originate from any early representation including: image data, early features used for building the saliency map, or the saliency map itself. The circuit for sensory attention described so far only receives and has access to the saliency map. However, with the suggested implementation, the local information from the saliency map can be easily transferred to the global processor. In fact, the magnitude of the localized feature in the saliency map is continuously reported to the global processor, as it is inherently measured by the WTA circuit. If the surrounding points are also needed, the global processor can program a trivial 1 x 1 active region at the desired location. The global processor inhibits all inputs of the saliency map except the programmed cell, and forces the WTA circuit to choose that particular point as the winner and report its magnitude on the global wire. We scanned the 1 x 1 active region throughout the array and collected several images (Figure 6).

Figure 6: Images from the tracking sensor (24x24 pixels).

4. Experimental Evaluation

Two tracking sensors prototypes — 1D and 2D —

have been built and tested for static and dynamic performance. The static performance has been tested on an early 1D prototype with 20 cells fabricated in 2μ CMOS technology. The findings have been reported earlier in [Brajovic and Kanade, 1994].

The temporal response of the WTA circuit is important when tracking moving features within dynamic saliency map. The dynamics of the circuit is a function of the parasitic capacitance at the input node *V1* comprising capacitance of the photo detector and capacitances of the gates and drains attached to this node. For a cell to win or lose this capacitance must be charged and discharged with the photocurrent. For average room illumination the photo currents are very small, much less then 1nA. Therefore, the WTA circuit in its original configuration is slow. To improve the dynamic performance of the WTA circuit, several measures can be taken: (1) increase the photo current, (2) decrease parasitic capacitance C, and (3) reduce the voltage swing on the capacitance *C*. A modified WTA cell that implements all three of these measures is shown in Figure 7. The photo transistor amplifies the photo current, T_3 isolates capacitance of the photo detector, and T_4 acts as a pull–up and limits the voltage swing.

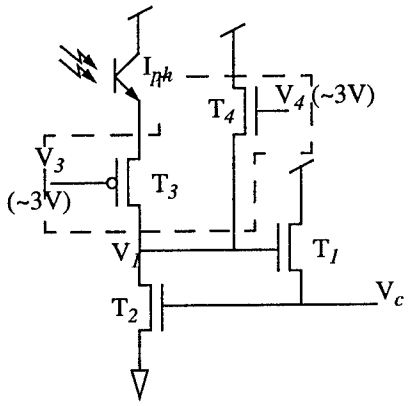


Figure 7: WTA cell with improved dynamic performance. (Fenced area indicates additions to the original WTA cell.)

The dynamic performance is evaluated for a 28x28–cell two–dimensional tracking computational sensor. Each cell is 62μ square. The photo transistor occupies about 30% of the cell's area. In the experimental set up, a scanning mirror reflects a beam of light onto a white cardboard. This produces a dot which travels along a straight line. The tracking sensor images the scene and tracks the moving dot. The rows of the sensor are approximately aligned with the trajectory of the laser dot, so that only x position needs to be observed. The mirror is driven from a sinusoidal oscillator whose frequency is adjustable. The maximum instantaneous velocity is attained at the middle of the trajectory. The goal is to observe how quickly the tracking sensor can shift attention, that is, how quickly it can update the feature's location as the feature travels across the array of cells. From the geometry of the set up, we can derive feature velocity from the frequency of the scanning mirror and then express it in image coordinates.

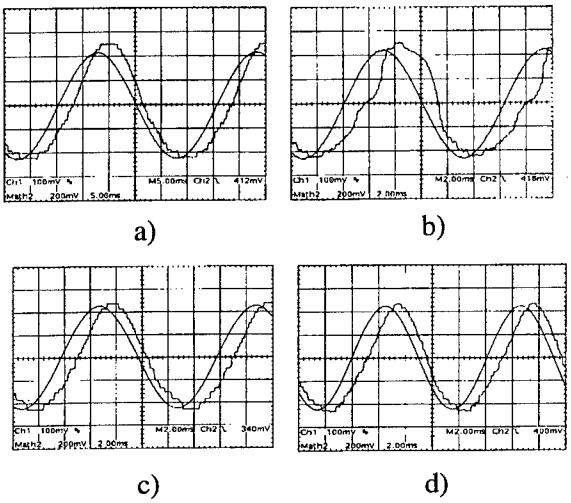


Figure 8: Tracking performance a) without the current buffer and without the pull–up. f=33Hz., b) without the current buffer and without the pull–up. f=83Hz., c) Tracking performance with the current buffer but without the pull–up. f=83Hz., d) Tracking performance with both the current buffer and

The effects of the current buffer and the pull–up can be turned on or off by biasing V_3 and V_4. Without the buffer and the pull–up, the sensor was reliably tracking up to the scanning frequency of 33Hz or 2,303.6 cells/second. Figure 8a shows two measured waveforms: (1) the feature's position x as reported by the tracking sensor, and (2) the sinusoid driving the mirror. If the frequency of the mirror is further increased, the reported position begins to distort. This is illustrated in Figure 8b for the scanning frequency of 83Hz. The tracking capability of the sensor starts to break down in the middle of the trajectory, as the velocity of the feature is the greatest there. Then, the current buffer is turned on by biasing V_3. The dynamic performance improved: the maximum tracking frequency is

increased from 33Hz to about 83.3Hz or from 2303.6 to 5793.9 cells/second. This is shown in Figure 8c; previously distorted waveform for the feature's position now better resembles the sinusoid. Finally, the pull up transistor is turned on by biasing V_4. The dynamic performance is slightly improved as shown in Figure 8d — the feature tracking is improved from 83Hz to about 100Hz, or to 6980.6 cells/second.

Another set of experiments is performed to evaluate how the intensity of the feature influences the dynamic performance. Using neutral density filters placed in front of the sensor's lens, the light is controllably attenuated. For each filter, the frequency of the mirror is increased until the waveform of the feature's position begins to distort. In this way, the maximum frequency is estimated for each intensity. Two sets of experiments are performed: (1) without the buffer and the pull–up, and (2) with the buffer and the pull–up. The results are graphed in Figure 9.

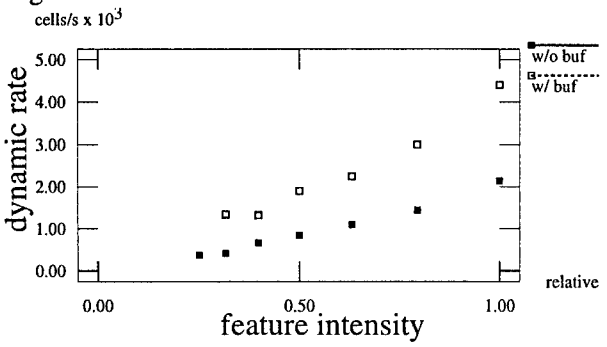


Figure 9: Maximum angular velocity of the attention shifts as a function of the relative feature intensity.

5. Conclusion

The proposed implementation for the sensory attention exhibits several interesting features. It performs a global operation over the saliency map and produces few global results: the position and magnitude of the selected saliency feature. These global results can be routed off–chip with low latency via few output pins. Furthermore, in the tracking mode, the global results are used internally for programming a 3 x 3 active region. This a top–down feedback secured robust performance in tracking the feature of interest while ignoring interference from other potentially stronger sources.

References

[Allport, 1989] Allport, A. "Visual Attention," Foundation of Cognitive Science, M. Posner (ed.), MIT Press, 1989, pp. 631–682.

[Andreou et al. 1992] A.G. Andreou, et al., "Current-Mode Subthreshold MOS Circuits for Analog VLSI Neural Systems," *IEEE Trans. on NN*, Vol. 2, No. 2, pp. 205-213, March 1992

[Brajovic and Kanade, 1994] Brajovic, V. and T. Kanade, "Computational Sensors for Global Operations," *IUS Proceedings*, pp. 621-630, 1994.

[Brajovic and Kanade, 1996] Brajovic, V. and T. Kanade, "A Sorting Image Sensor: An Example of Massively Parallel Intensity-to-Time Processing for Low-Latency Computational Sensors," Proc. of the 1996 IEEE Intl. Conf. on Robotics and Automation, April 1996, Minneapolis, MN.

[DeWeerth, 1992] DeWeerth, S.P., "Analog VLSI Circuits for Stimulus Localization and Centroid Computation," *Intl. Jour. of Comp. Vision*, Vol. 8, No. 3, 1992, pp. 191-202.

[Horn, 1986] Horn, B., *Robot Vision*, MIT Press, 1986.

[Kanade and Bajcsy, 1993] Kanade, T. and R. Bajcsy, "Computational Sensors: A Report from DARPA workshop", *IUS Proceedings*, 1993.

[Koch and Ullman, 1987] Koch, C. and S. Ullman, "Shifts in Selective Visual Attention: Toward the Underlying Neural Circuitry. In L.M. Vaina (edt.), *Matters of Intelligence*, Reidel Publishing, 1987, pp. 115-141.

[Lazzaro et al., 1988] J. Lazzaro, S. Ryckebusch, M.A. Mahowald and C. Mead, "Winner-Take-All Networks of O(n) Complexity," in *Adv. in Neural Inf. Proc. Sys. Vol. 1*, D. Tourestzky, ed., pp. 703-711, Morgan Kaufmann, San Mateo, CA, 1988.

[Milanese, 1993] R. Milanese, *"Detecting Salient regions in an Image: From Biological Evidence to Computer Implementation,"* Ph.D., Dept. of Com. Sci., U. of Genova, Switzerland, Dec. 1993.

[Morris and DeWeerth, 1996] T.G. Morris and S.P. DeWeerth, *Analog VLSI Circuits for Covert Attentional Shifts*, MicroNeuro 1996, Lausanne, Switzerland.

Experiments with an Algorithm for Recovering Fluid Flow from Video Imagery

R. P. Wildes, M. J. Amabile, A. M. Lanzillotto and T. S. Leu*
David Sarnoff Research Center, Inc.
Princeton, NJ 08543
E-MAIL: {wildes,mja,aml,jleu}@sarnoff.com

Abstract

A physics-based algorithm for recovering fluid flow from video imagery has been developed. The physical basis comes about as constraints from fluid mechanics are folded into an optical flow algorithm. A series of empirical studies are presented that evaluate the performance of this algorithm in the face of both synthetic and natural imagery. In these experiments, the fluids are seeded with tracer particles so that the flow is visible. For cases where the expected flow can be predicted analytically, the recovered velocity fields are in good accord with theory. For complex flows, where the results cannot be predicted analytically, the recovered velocity fields agree with qualitative expectations.

1 Introduction

Constraints derived from physical considerations are often used to provide the basis for well motivated computer vision algorithms. One domain where this methodology is likely to be of value is the measurement of fluid flow from video imagery. Here, applicable physical constraints come naturally from fluid mechanics. Further, it is standard practice in experimental fluid mechanics to seed flows with tracer particles for the sake of visualization and analysis. Video of these flows is rich in image detail and therefore well suited to computer vision techniques. Following this motivation, a physics-based algorithm for recovering fluid flow from video imagery has been developed. This paper presents the results of a series of empirical studies that illustrate the algorithm's performance.

The recovery of optical flow, i.e., the apparent motion of image intensity patterns, has been the subject of a great deal of research [Beuchemin and Barron, 1995]. Typically, optical flow algorithms are based on the brightness constancy constraint [Horn, 1986]. This constraint dictates that the algorithms establish a mapping between two images based directly on the similarities of image intensities. Most closely related to the algorithm that is the subject of the current paper are previous approaches that employed constraints based on fluid flow continuity equations [Del Bimbo *et al.*, 1993, Fitzpatrick, 1995, Schunk, 1986, Song and Leahy, 1991]. Surprisingly, this body of research did not apply the resulting algorithms to the domain of fluid flow recovery.

Previous research from the computer vision community that has been concerned with the recovery of fluid flow from images has not made direct use of constraints derived from fluid mechanics [Jahne and Waas, 1989, Maas *et al.*, 1994]. Investigators from fluid mechanics also have developed approaches to making flow measurements from images. Methods in that domain have concentrated on image correlation techniques (occasionally with extensions) or simple particle trackers [Adrian, 1991].

*The research described in this paper was supported by DARPA/ETO under contract No. DABT63-95-C-0057.

2 Description of algorithm

Let $E(x,y,t)$ be an image, a function of spatial coordinates, (x,y), and time, t. Suppose that this image depicts a fluid flow in such a way that the essential physical characteristics of the flow are captured. In particular, suppose that the imaged intensities observe the conservation of mass just as does the fluid density. For example, the 2D transmittance image of a 3D fluid flow that respects conservation of mass in 3D is a 2D flow that respects the conservation of mass in 2D [Wildes *et al.*, 1997]. This is true provided that there is no material loss due to normal flow at the boundaries of the flow. In this case the 2D (imaged) flow is the density weighted average of the 3D flow, taken along imaging rays. Correspondingly, application of the conservation of mass flow continuity equation to a temporally varying image yields

$$\nabla \cdot (E\mathbf{v}) + E_t = 0 \tag{1}$$

where $\mathbf{v} = (u(x,y,t), v(x,y,t))$ is the imaged flow, $\nabla \equiv (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ is the spatial gradient operator and subscripts denote partial differentiation [Streeter and Wylie, 1985]. This continuity equation will be taken as the fundamental constraint for relating image intensity measurements to fluid flow. In practice, to allow for imperfect data, strict enforcement of the continuity constraint is replaced with minimization of

$$c_c = (\nabla \cdot (E\mathbf{v}) + E_t)^2 \tag{2}$$

with respect to $\mathbf{v}$ over an image domain of interest.

To ameliorate the effects of noise, a second constraint is imposed that encourages smoothness in the recovered flow. This notion is captured by minimizing the spatial variation of the flow, i.e.,

$$c_s = u_x^2 + u_y^2 + v_x^2 + v_y^2. \tag{3}$$

Without imposition of this constraint, it was found that imagery of interest was too noisy to yield coherent flow fields.

Following [Horn and Schunk, 1981], the measures of continuity (2) and smoothness (3) can be combined and evaluated over a domain of interest to yield a problem of the form

$$\min \iint (\lambda c_c + c_s)\, dx\, dy, \tag{4}$$

with λ a weighting parameter that trades off adherence to the continuity constraint and smoothness of flow. The variational calculus [Courant and Hilbert, 1953] can be applied to this integrated constraint equation to derive necessary conditions for a minimum with respect to flow parameters, (u,v). The corresponding (Euler) partial differential equations relate image intensity measurements to permissible flows according to

$$\begin{aligned} \nabla^2 u &= -\lambda(E_{tx} + E_{xx}u + E_{yx}v + 2E_x u_x \\ &\quad + E_y v_x + E_x v_y + E v_{yx} + E u_{xx})E \\ \nabla^2 v &= -\lambda(E_{ty} + E_{xy}u + E_{yy}v + E_y u_x \\ &\quad + E_x u_y + 2E_y v_y + E u_{xy} + E v_{yy})E, \end{aligned} \tag{5}$$

where $\nabla^2 \equiv (\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})$ is the Laplacian operator. Boundary conditions for these equations can be had via further appeal to physical principles. Along the edges of channels that contain the fluid, the flow is constrained to be tangential to the channel walls; if the channel walls do not completely enclose the domain of interest (e.g., the apparent end of a channel as it runs off a side of the image), then natural boundary conditions are enforced. These conditions are appropriate for the flows studied in this paper; others might be required in different situations.

A discrete, iterative solution to the Euler equations, (5), can be had by letting $\mathcal{I}$ be the identity matrix,

$$\mathcal{A} = E \begin{pmatrix} E_{xx} - E & E_{yx} \\ E_{xy} & E_{yy} - E \end{pmatrix},$$

$$\mathbf{b} = -E \begin{pmatrix} E_{tx} + 2E_x \Delta_x u_{i,j} + E_y \Delta_x v_{i,j} + \\ E_{ty} + E_y \Delta_x u_{i,j} + E_x \Delta_y u_{i,j} + \\ E_x \Delta_y v_{i,j} + E \Delta_{yx} v_{i,j} + E \bar{u}^x_{i,j} \\ 2E_y \Delta_y v_{i,j} + E \Delta_{xy} u_{i,j} + E \bar{v}^y_{i,j} \end{pmatrix},$$

with i,j image position indices, $\Delta_x u_{i,j} = (u_{i+1,j} - u_{i-1,j})/2$ the central difference, $\Delta_{xy} u_{i,j} = (u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1})/4$ the mixed difference, while $\bar{u}^x_{i,j} =$

$(u_{i+1,j} + u_{i-1,j})/2$ and $\bar{u}_{i,j} = (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1})/4$ are averaging operators, etc., to arrive at

$$\mathbf{v}^{n+1} = (\mathcal{I} + \lambda\mathcal{A})^{\dagger}(\bar{\mathbf{v}}^{n} + \lambda\mathbf{b}), \qquad (6)$$

where $\mathbf{v}^{n+1}$ is the value of $\mathbf{v} = (u, v)^{\top}$ computed at iteration $n+1$ from the value $\mathbf{v}^{n}$ computed at iteration n and $\dagger$ denotes the matrix inverse. Further details on this derivation can be found in a companion paper [Wildes *et al.*, 1997].

The iterative solution for flow field components (6) has been embodied in a Gauss-Seidel relaxation algorithm [Dahlquist and Bjork, 1974]. This algorithm accepts a pair of images as well as a region of interest map and recovers a corresponding flow field. The initial values for u and v are taken as zero everywhere. For the sake of computational efficiency, this algorithm has been embedded in a hierarchical coarse-to-fine refinement control structure [Bergen *et al.*, 1991]. In the following experiments, coarse-to-fine processing proceeded up to five levels of resolution reduction. Algorithm iterations were calculated to allow the most distant pixels at the coarsest resolution to broadcast their information to one another.

3 Empirical studies

Two classes of empirical studies have been conducted. First, synthetic imagery has been used to evaluate the algorithm in the face of known "ground truth". Second, natural imagery has been used to evaluate the algorithm with real world data. Since the current target application for the algorithm is microfluidics, all studies have been executed at microscopic scales. Nevertheless, the algorithm should exhibit similar performance at macroscopic scales.

Synthetic images were generated to simulate the transmittance imagery that has served as the major source of empirical data to date. The basic experimental preparation consists of steady state fluid flow through cylindrical tubes. This preparation is of interest because the expected flow can be predicted analytically according to the fully developed circular pipe flow model [Schichting, 1979]; therefore, recovered flows can be evaluated against model predictions. The pipe flow model dictates a parabolic displacement along the axis of the tube with the form

$$\frac{v}{v_{max}} = 1 - \frac{r^2}{R^2} \qquad (7)$$

where R is the tube radius, r is the perpendicular distance of any point in the tube from the central axis, v_{max} is the maximal displacement along the central axis and v is displacement along the tube axis at point r. Flow in the orthogonal direction is taken as zero. To make the flow visible, the fluids are composed of mixtures of liquids in the form of tiny droplets that result in spatially varying x-ray absorption. Image sequences of this device are captured via microradiography [Cosslett *et al.*, 1957].

To mimic this set-up, spheres were generated and randomly dispersed in a cylinder. The density and diameters of the spheres were chosen to be in accord with real experiments. A ray-tracer was used to simulate the transmission of x-rays through these structures according to a standard linear absorption model [Barrett and Swindell, 1981]. The simulated spatial resolution was 2.8 microns/pixel. The grey-level resolution was 16 bits. A second image was ray-traced after shifting the spheres according to the pipe flow model (7). The left panel of Figure 1 shows an image from a simulation experiment.

Simulated flow sequences were generated for tube diameters of 1000, 800 and 600 microns. For each tube, flow rates were simulated to yield a range of maximum image displacements. The flow recovery algorithm was executed on the resulting image sequences. In these experiments $\lambda = 0.01$, an empirically selected value. To quantify performance, the root mean square error was calculated between the recovered and simulated velocities in the direction of the tube axis. (The recovered velocities in the orthogonal direction were inconsequential.) The results are shown in the right panel of Figure 1. For small displacements the error is small for all tubes. With increased displacement the error rises, especially for smaller tubes. While not apparent in these plots, the error typically comes as an underestimate of the true flow. The errors are due to the fact that the algorithm requires more

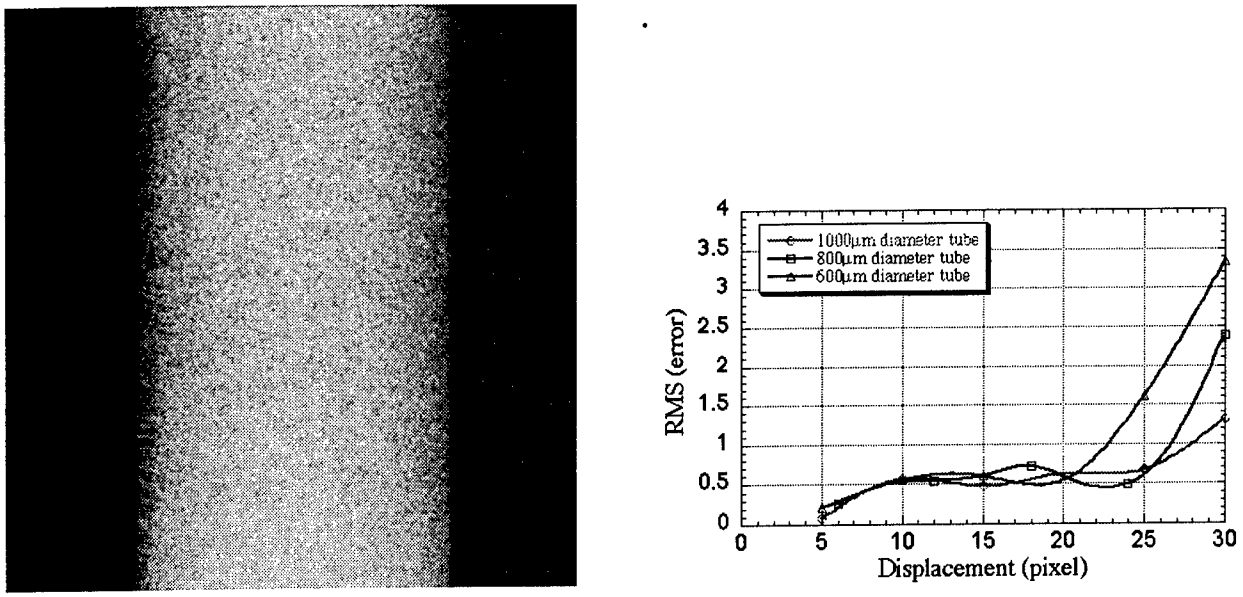


Figure 1: Root Mean Square (RMS) Error of Flow Recovered from Synthetic Imagery. The left panel shows a synthetic radiograph of emulsion particles flowing through a capillary tube. The right panel shows the RMS error of recovered vs. veridical displacements.

spatial support in the recovery of larger displacements. The narrower tubes might not offer sufficient information to support the recovery of large displacements.

Experiments with natural imagery involved flow sequences generated via microradiography or visible light microscopy of fluid emulsions driven through a variety of devices. The emulsions consisted of a contrast medium mixed with a fluid. After emulsification, the contrast medium was dispersed in the fluid as tiny droplets (1-20 microns). By choosing a contrast medium with the same density as the fluid the droplets were made neutrally buoyant and followed the imposed flow. The flow was generated by a mechanically driven syringe pump that forced the fluid emulsion through the devices.

The first series of natural image experiments involved microradiography of steady state flow through cylindrical quartz tubes. Images were generated by a collimated monochromatic x-ray source in conjunction with a phosphor screen and optics to image onto a CCD imager for digitization at 14 bit precision. Flow sequences were generated for tube diameters of 1000, 840, 750 and 640 microns. For each tube, flows were injected at two rates, 0.004 and 0.008 microliters per second. The spatial resolution was 2.8 microns/pixel with a temporal sampling rate of 0.4 frames/second. and an exposure time of 500 milliseconds/frame. An image from an experiment with a 1000 micron tube and flow rate of 0.004 microliters/second is shown in the upper left panel of Figure 2.

The flow recovery algorithm was executed on all of the resulting image sequences. In these experiments $\lambda = 0.0001$, an empirically selected value. The recovered flow for the 1000 micron tube at the lower flow rate is shown as a vector plot in the upper right panel of Figure 2. The average recovered velocity profiles in the direction of the tube axis are shown for the 840 micron tube in the lower left panel of Figure 2. For comparison, the profiles predicted by fully developed pipe flow (7) averaged along imaging rays is plotted in the same figure. (Recall that the imaged flow should be an average of the 3D flow due to the properties of transmittance imaging.) The recovered flow is in good agreement with the predictions of the model for both flow rates. The lower right panel of Figure 2 shows the recovered flow profiles for the 1000, 840, 750 and 640 micron tubes collapsed into a single non-dimensional plot. The radius of the

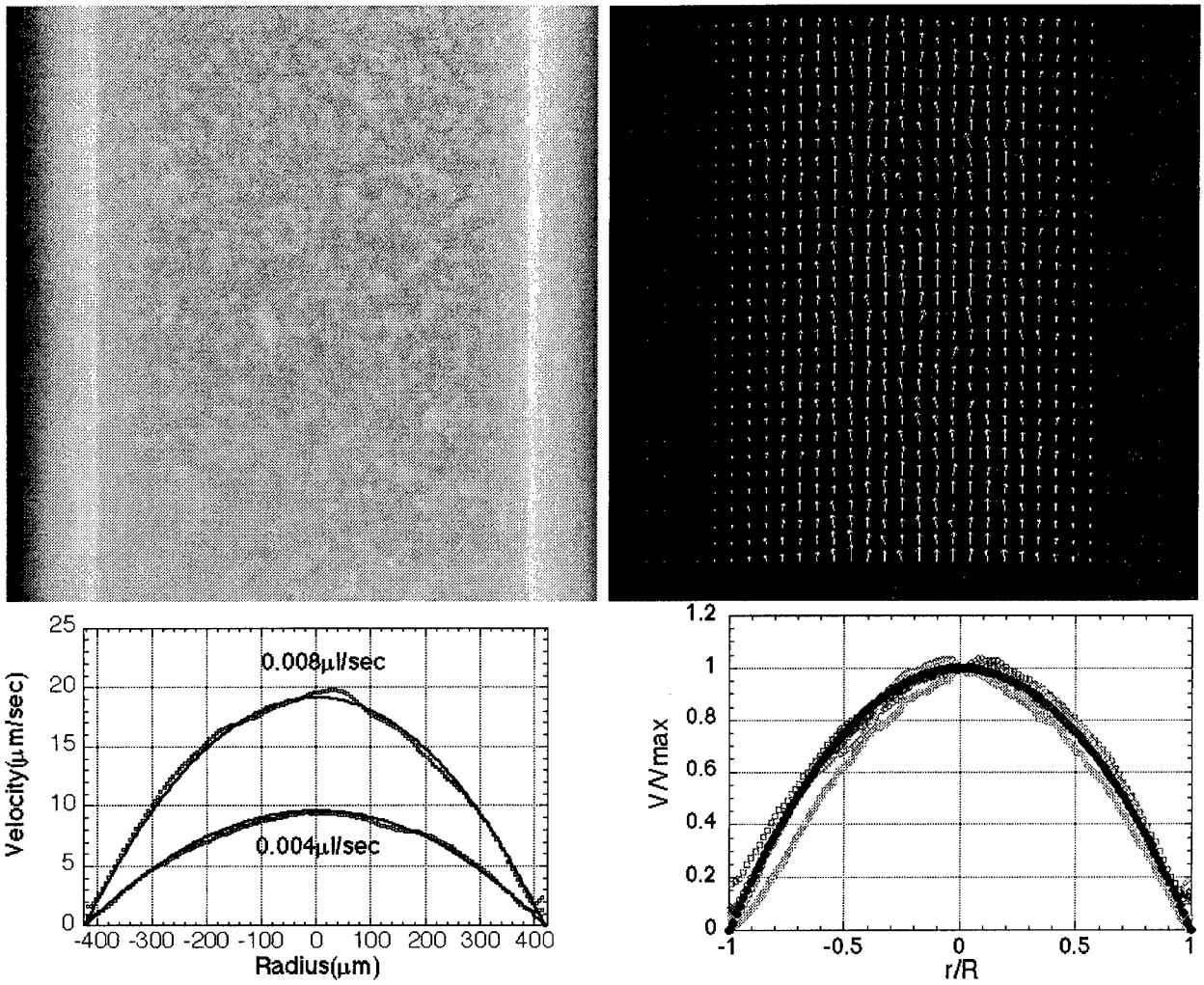


Figure 2: Velocity Profiles Recovered from Natural Imagery of Fluid Flow through Capillary Tubes. The upper left panel shows a frame from a radiograph sequence of an emulsion flowing through a capillary tube. The upper right panel shows the recovered velocity field from two successive frames. The lower left panel shows the average recovered vertical flow profiles through an 800 micron capillary tube for two different flow rates. The corresponding analytically predicted flow profiles for these experiments also are shown (solid line). The lower right panel shows in dimensionless coordinates the average recovered vertical flow profiles for a range of tube diameters and flow rates. The corresponding analytically predicted flow profile for these experiments also is shown (solid line).

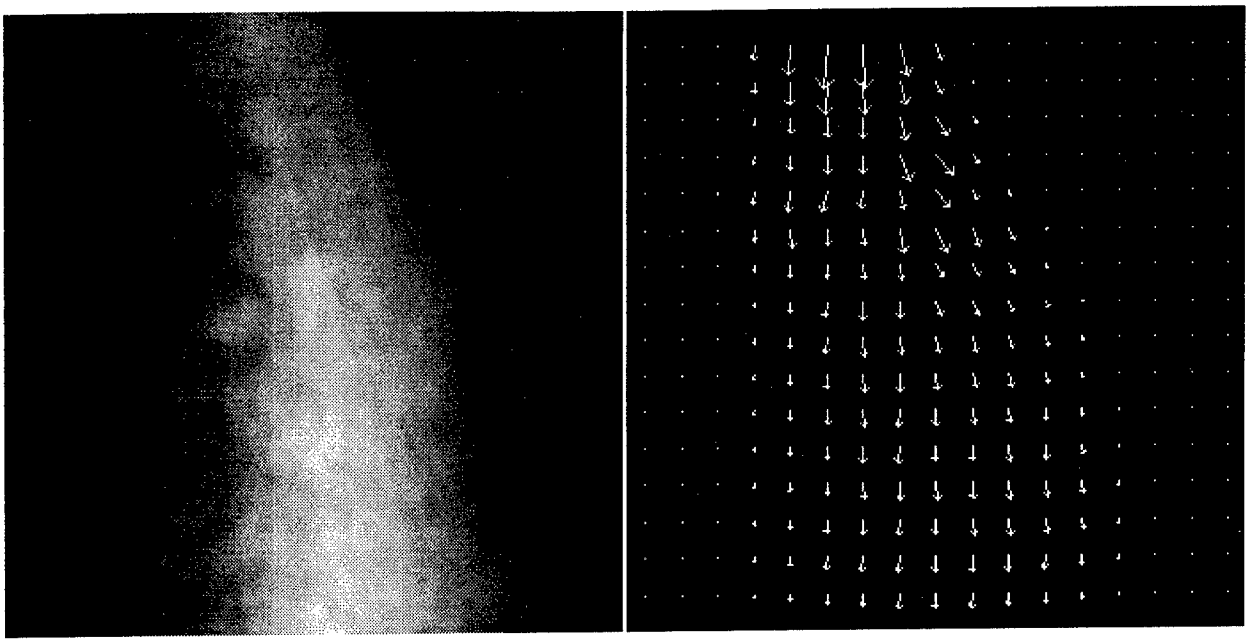


Figure 3: Velocity Field Recovered from Natural Imagery of Fluid Flow through a Ramped Step Channel. The left panel shows a frame from a radiograph sequence of an emulsion flowing through a ramped step channel etched in silicon and covered with Pyrex glass. The right panel shows the recovered velocity field from two successive frames. The recovered flow is in accord with one's visual impression upon viewing the image sequence.

capillary tube, R, and the maximum velocity at centerline, v_{max}, are used to non-dimensionalize the plot. A pipe flow velocity profile also is superimposed on this plot. In all cases, the recovered profile is in good agreement with the theoretical prediction.

A second set of studies involved microradiography of more complicated devices where analytic predictions of the flow were not available. The first device was a ramped step channel. The second device was a serpentine channel. Both devices were etched in silicon and covered with Pyrex glass. Parameters for both studies were the same: Injected flows were 0.004 microliters/second; spatial resolution was 1.6 microns/pixel with temporal sampling 10 frames/second and exposure time of 100 milliseconds/frame. $\lambda = 0.00001$, an empirically selected value. An image of the ramped step channel and a recovered velocity field are shown in Figure 3. The results are in accord with one's visual impression: The flow is fastest at the upper inlet and looses speed while expanding thereafter; the speed is smallest near the channel walls. An image of the serpentine channel and a recovered velocity field are shown in Figure 4. Again, the results are in accord with one's visual impression: The flow follows the channel's bends most closely along the boundaries and less so in the center; the speed is smallest near the channel walls.

The final study returned to the circular pipe flow, but acquired with visible light microscopy. In this case the flow was injected at 0.08 microliters/second. Focus was used to isolate a thin layer of fluid flow for capture with a CCD camera and 8 bit digitization. The spatial resolution was 3.9 microns/pixel with temporal sampling at 30 frames/second and an exposure time of 33 milliseconds/frame. $\lambda = 0.01$, an empirically selected value. An image from this experiment and a recovered velocity field are shown in Figure 5. A lack of precision in focus precludes application of the pipe flow model (7) to yield an analytic prediction for comparison. However, the results are in accord with one's visual impression and they exhibit the parabolic flow profile characteristic of this geometry.

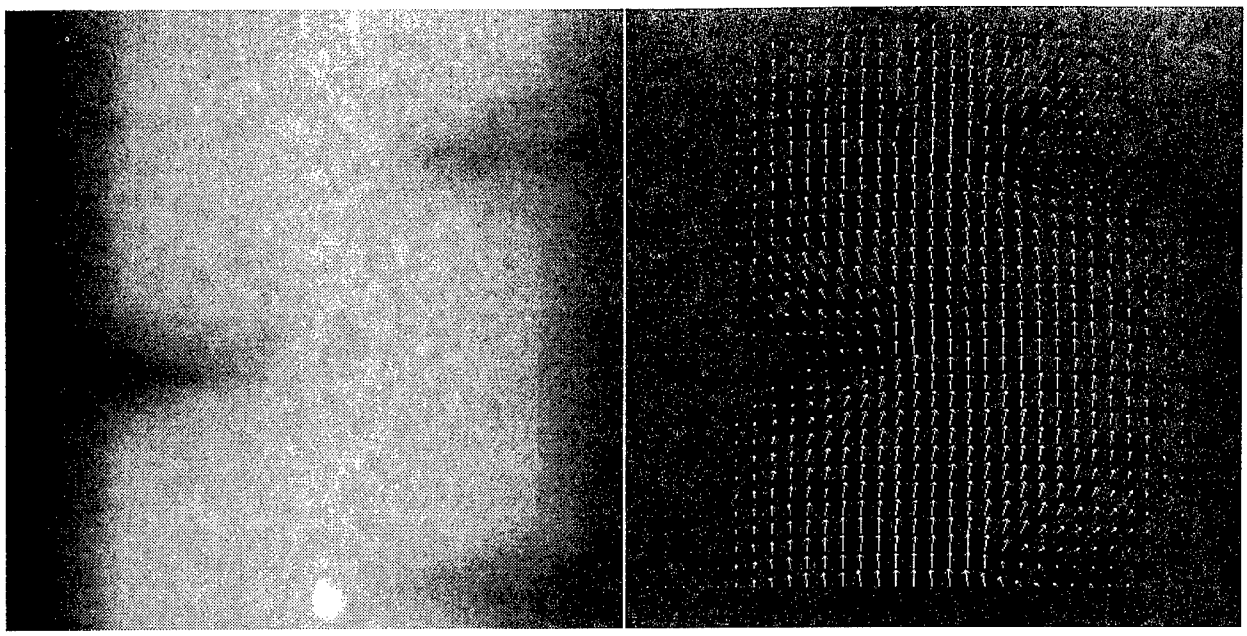


Figure 4: Velocity Field Recovered from Natural Imagery of Fluid Flow through a Serpentine Channel. The left panel shows a frame from a radiograph sequence of an emulsion flowing through a serpentine channel etched in silicon and covered with Pyrex glass. The right panel shows the recovered velocity field from two successive frames. The recovered flow is in accord with one's visual impression upon viewing the image sequence.

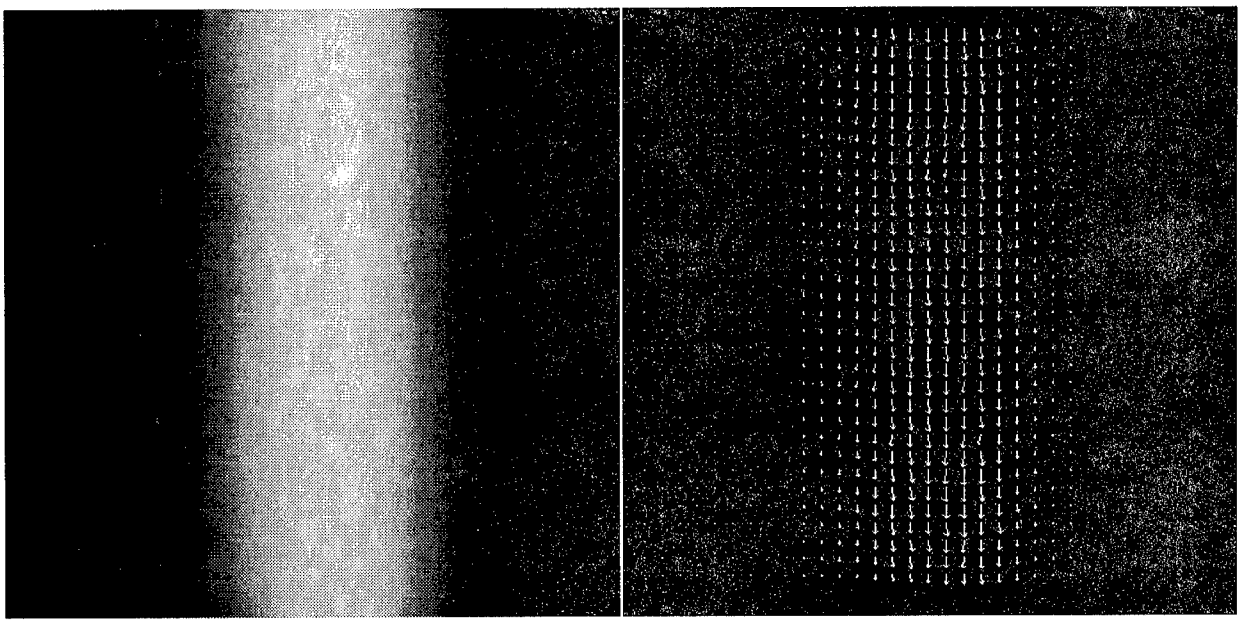


Figure 5: Velocity Field Recovered from Natural Imagery of Fluid Flow through a Capillary Tube. The left panel shows a frame from a reflected light image sequence of an emulsion flowing through a capillary tube. The right panel shows the recovered velocity field from two successive frames. The recovered flow is in accord with one's visual impression upon viewing the image sequence.

4 Summary

This paper has presented a series of empirical studies with a computer vision algorithm for recovering fluid flow from video imagery. The algorithm is based on physical principles derived from fluid mechanics. Testing has made use of both synthetic and natural image sequences that depict fluids containing a contrast medium flowing through a variety of devices. For cases where the flow can be modeled analytically, the recovered velocity fields are in good agreement with predictions. For more complex flows, where analytic predictions cannot be made, the recovered velocity fields are in accord with qualitative expectations.

Acknowledgments

J. Dunsmuir provided assistance in acquiring the microradiographs.

References

[Adrian, 1991] R. J. Adrian. Particle imaging techniques for experimental fluid mechanics. *Annual Review of Fluid Mechanics*, 23:361–304, 1991.

[Barrett and Swindell, 1981] H. H. Barrett and W. Swindell. *Radiological Imaging.* Academic Press, London, 1981.

[Bergen *et al.*, 1991] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proc. ECCV*, pages 5–10, 1991.

[Beuchemin and Barron, 1995] S. S. Beuchemin and J. L. Barron. The computation of optical flow. *ACM Computing Reviews*, 27(3):433–467, 1995.

[Cosslett *et al.*, 1957] V. E. Cosslett, A. Engstrom, and H. H. Pattee. *X-Ray Microscopy and Microradiography.* Academic Press, NY, NY, 1957.

[Courant and Hilbert, 1953] R. Courant and D. Hilbert. *Methods of Mathematical Physics.* John Wiley & Sons, NY, NY, 1953.

[Dahlquist and Bjork, 1974] G. Dahlquist and A. Bjork. *Numerical Methods.* Prentice Hall, Englewood Cliffs, New Jersey, 1974.

[Del Bimbo *et al.*, 1993] A. Del Bimbo, P. Nesi, and J. Sanz. Optical flow estimation by using classical and extended constraints. In *Proc. Int. Workshop Time-Varying Image Processing*, pages 351–358, 1993.

[Fitzpatrick, 1995] J. M. Fitzpatrick. A method for calculating velocity in time dependent images based on the continuity equation. In *Proc. CVPR*, pages 78–81, 1995.

[Horn and Schunk, 1981] B. K. P. Horn and B. G. Schunk. Determining optical flow. *AI*, 17(1–3):185–203, 1981.

[Horn, 1986] B. K. P. Horn. *Robot Vision.* MIT Press, Cambridge, MA, 1986.

[Jahne and Waas, 1989] B. Jahne and S. Waas. Optical wave measuring technique for small scale water surface waves. In *Proc. Adv. Opt. Instr. Rem. Sens.*, pages 147–152, 1989.

[Maas *et al.*, 1994] H. Maas, A. Stefanidis, and A. Gruen. Feature tracking in 3-D fluid tomography sequences. In *Proc. ICIP*, pages 530–534, 1994.

[Schichting, 1979] H. Schichting. *Boundary Layer Theory.* McGraw-Hill, NY, NY, 1979.

[Schunk, 1986] B. G. Schunk. Image flow continuity equations for motion and density. In *Proc. IEEE Workshop on Motion*, pages 89–94, 1986.

[Song and Leahy, 1991] S. M. Song and R. M. Leahy. Computation of 3-D velocity fields from 3-D cine and ct images of a human heart. *IEEE MI*, 10(3):295–306, 1991.

[Streeter and Wylie, 1985] V. L. Streeter and E. B. Wylie. *Fluid Mechanics.* McGraw-Hill, NY, NY, 1985.

[Wildes *et al.*, 1997] R. P. Wildes, M. J. Amabile, A. M. Lanzillotto, and T. S. Leu. Physically based fluid flow recovery from image sequences. In *Proc. CVPR*, 1997. to appear.

Real-Time 3-D Tracking and Classification of Human Behavior

Alex Pentland, Ali Azarbayjani, Nuria Oliver, Matt Brand
Perceptual Computing Section, MIT Media Laboratory, Cambridge, MA, USA

Abstract

We describe a system for real-time 3-D estimation and classification of human behavior using only modest computational resources. The system is based on use of 2-D blob features, *which are clusters of similar pixels in the image plane and can arise from similarity of color, texture, motion and other signal-based metrics. We use nonlinear modeling and a combination of iterative and recursive estimation methods to recover 3-D geometry from blob correspondences across multiple images. The 3-D geometry includes the 3-D shapes, translations, and orientations of blobs and the relative orientation of the cameras. The system is self-calibrating and can track people's head and hands with RMS errors of 1–2 cm in translation and 2 degrees in rotation.*

Patterns of behavior (e.g., hand or face gestures) can then be classified in real-time using Hidden Markov Model (HMM) methods, importantly including the new Coupled HMM methods that we have recently developed (Brand, Oliver and Pentland 1997). Typical classification accuracies are near 100%.

1 Introduction

This paper describes a real-time, self-calibrating system for accurate 3-D person tracking using 2-D *blob features*. The 3-D and 2-D output of the tracking is then used to reliably classify a wide variety of hand and face gestures, by use of either traditional Hidden Markov Models (HMMs) and our new Coupled HMMs. All of the experimental apparatus described here is real-time, at 20 to 30 frames per second, and runs on low-end SGI workstations without any special-purpose hardware.

The notion of "blobs" as a representation for image features has a long history in computer vision [4, 5, 3, 9]. The term "blob" is somewhat self-explanatory ("something of vague or indefinite form"), but a useful definition from a computational point of view might be that a blob is defined by some visual property that is shared by all the pixels contained in the blob and is not shared by surrounding pixels. This property could be color, texture, brightness, motion, shading, a combination of these, or any other salient spatio-temporal property derived from the signal (the image sequence).

Our current interest in blob models is motivated by our discovery that they can be reliably tracked even in complex, dynamic scenes, and that they can be extracted in real-time without the need for special purpose hardware. These properties are particularly important in applications that require tracking people, and recently we have used 2-D blob tracking for real-time whole-body human interfaces [9] and real-time recognition of American Sign Language hand gestures [8].

Our success at 2-D tracking motivates this paper's investigation into recovering useful 3-D geometry from these features. We begin by addressing the basic mathematical problem of estimating 3-D geometry from blob correspondences in displaced cameras. The relevant unknown 3-D geometry includes the shapes and motion of 3-D objects and the relative orientation of the cameras. The observations consist of the corresponding 2-D blobs, which can in general be derived from any signal-based similarity metric; in our experiments we use chrominance spectral similarity.

We will then present experimental results on various sub-problems required for human behavior interpretation, including self-calibrating the stereo rig, 3-D estimation of hand/head shape and motion, and recognition of hand and face gestures. Additional detail on the 3-D estimation can be found in Azerbayejani and Pentland [1], on the mouth tracking and recognition in Oliver, Bernard, Coutaz, and Pentland [6], and on the coupled HMM formulation in Brand, Oliver, and Pentland [2].

2 Background: Blob features

The notion of grouping atomic parts of a scene together to form blob-like entities based on proximity and visual appearance is a natural one, and has been of interest to visual scientists since the Gestalt psychologists studied grouping criteria early in this century.

In modern computer vision processing we seek to group pixels of images together and to "segment" images based on visual coherence, but the "features" obtained from such efforts are usually taken to be the boundaries, or contours, of these regions rather than the regions themselves. In very complex scenes, such as those containing people or natural objects, contour features often prove unreliable and difficult to find and use.

The blob representation that we use was developed by Kauth *et al* and Pentland [7, 4], for application to multispectral satillite (MSS) imagery. In this method feature vectors at each pixel are formed by adding (x, y) spatial coordinates to the spectral components of the imagery. These are then clustered so that color and spatial similarity combine to form coherent connected regions or "blobs". Essentially the same technique has been used recently in Wren *et al* [9] for real-time tracking of people in color

video. The spatial coordinates are combined with color and brightness channels to form a five-element feature vector at each point (x, y, Y, U, V). These are clustered into blobs which drive a "connected-blob" representation of the person.

By using Expectation Maximization (EM) methods to obtain Gaussian mixture models for the spatio-chrominance feature vector, very complex shapes and color patterns can be adaptively estimated from the image stream. In our system we use an incremental version of EM, which allows us to adaptively and continuously update the spatio-chromatic blob descriptions. Thus not only can we adapt to very different skin colors, etc., but also to changes in illumination.

3 Estimating 3-D Geometry

We can represent shapes in both 2-D and 3-D by their low-order moments. Clusters of 2-D points have 2-D spatial means and covariance matrices, which we shall denote $\bar{q}$ and C_q, while 3-D shapes have 3-D spatial means and covariance matrices, denoted $\bar{p}$ and C_p.

The blob statistics can be interpreted as representing uniform, Gaussian, or some other second-order distribution of occupancy. The distribution is not terribly important because the physical location and orientation are independent of distribution and are encoded in the mean and covariance. The scale of the shape parameters with respect to the variance will vary across distributions, but the relative shape will remain the same. A Gaussian interpretation is chosen, therefore, for computational convenience.

Like other representations used in computer vision and signal analysis, including superquadrics, modal analysis, and eigen-representations, blobs represent the global aspects of the shape and can be augmented with further moments to attain more detail if the data supports it. The reduction of degrees of freedom from individual pixels to blob parameters is a form of regularization which allows the ill-conditioned problem to be solved in a principled and stable way.

For both 2-D and 3-D blobs, there is a useful physical interpretation of the blob parameters. The mean represents the geometric center of the blob area (2-D) or volume (3-D). The covariance, being symmetric, can be diagonalized via an eigenvalue decomposition

$$C = \Phi L \Phi^T \tag{1}$$

where Φ is orthonormal and L is diagonal.

The diagonal L matrix represents the size of the blob along independent orthogonal object-centered axes and Φ is a rotation matrix that brings this object-centered basis in alignment with the coordinate basis of C.

This decomposition and physical interpretation is important for estimation, because the shape L is constant (or slowly varying) while the rotation Φ is dynamic. The parameters must be separated so they can be treated appropriately.

3.1 Parameterization

To estimate 3-D geometry from 2-D images, we consider the nonlinear forward model and try to solve the inverse problem. That is, we consider the function

$$y = f(x) \tag{2}$$

where y consists of 2-D observations and x consists of the 3-D state. We observe y and try to recover x.

The observation vector y consists of a $(\bar{q}, C_q)$ pair for each observation:

$$y = \{(\bar{q}, C_q)_{t,k}\}, t = 1 \ldots M, k = 1 \ldots N \tag{3}$$

where t is the frame index and k is the blob index.

The covariance matrix C_q has three free parameters, thus any pertubation δy is of dimension $5MN$ where M is the number of blobs and N is the number of images being observed ($N = 2$ for stereo, e.g). Thus, the corresponding observation perturbation is

$$\begin{aligned} \delta y &= \{(\delta\bar{q}_u, \delta\bar{q}_v, \delta\sigma_u^2, \delta\sigma_v^2, \delta\sigma_{uv})_{t,k}\} \\ &\quad t = 1 \ldots M, k = 1 \ldots N \end{aligned} \tag{4}$$

The state vector x consists of the 3-D blob parameters $(\bar{p}, C_p)$ and the 3-D transformation (T, R, β) to each camera, where T is a 3-D translation vector, R is a 3-D rotation (unit quaternion), and β is inverse effective focal length $(1/f)$. In order to facilitate the physical interpretation, the 3-D covariance is decomposed as in Equation 1. Thus, for M blobs and N cameras, x is

$$x = \{(\bar{p}, \Phi, L)_k, (T, R, \beta)_t\}, t = 1 \ldots M, k = 1 \ldots N \tag{5}$$

and the corresponding state perturbation is

$$\begin{aligned} \delta x &= \{(\delta\bar{p}_x, \delta\bar{p}_y, \delta\bar{p}_z, \delta\omega_x, \delta\omega_y, \delta\omega_z, \delta l_x, \delta l_y, \delta l_z)_k, \\ &\quad (\delta t_x, \delta t_y, \delta t_z, \delta\omega_x, \delta\omega_y, \delta\omega_z, \delta\beta)_t\}, \\ &\quad t = 1 \ldots M, k = 1 \ldots N \end{aligned} \tag{6}$$

where the $\delta\omega$ terms are angular perturbations on the respective rotations and the δl terms are perturbations on the elements of the diagonal matrix L.

3.2 Forward process model

The forward model begins with the perspective projection equation:

$$q = c(p) \tag{7}$$

$$\begin{pmatrix} q_u \\ q_v \end{pmatrix} = \frac{1}{1 + p_z\beta} \begin{pmatrix} p_x \\ p_y \end{pmatrix} \tag{8}$$

where p is a 3-D point, q is the corresponding image point, the origin is at the center of the image plane, and the center of projection is at coordinates $(0, 0, -1/\beta)$.

Three-dimensional Gaussian distributions do not perspectively project exactly to 2-D Gaussian distributions, but if we have a rough estimate of the 3-D mean, we can use the linearized projection equation which does project 3-D Gaussians to 2-D Gaussians. It is easy to get a good initial value p_0 for the 3-D mean from the 2-D observations $\{\bar{q}_i\}$ to facilitate this linearization; this is covered in the next section.

(Possible drawbacks to using the linearized projection equation are that it will introduce a systematic modeling error in the results and that it requires

iterative re-computation of p_0. However, the modeling error is small and most applications, particularly the human interface applications we are interested in, do not need higher precision. Our experimental results and applications developed using our system support this position. With regards to calculating p_0, it is an extremely cheap computation that we currently solve using a linear technique.)

Here we assume we have computed p_0 and linearize the projection equation

$$q = c(p_0) + J_c(x)(p - p_0) + \cdots \quad (9)$$

where J_c is the Jacobian matrix of partial derivatives $(\partial c/\partial \delta p)$.

With the linearized system the 3-D mean projects to the 2-D mean, thus we have

$$\bar{q} = c(\bar{p}) \quad (10)$$

To get the covariance observations, define $\delta q = q - \bar{q}$ and $\delta p = p - \bar{p}$. Since $\bar{q}$ is $c(\bar{p})$ we have the linear relationship

$$\delta q = J_c \delta p \quad (11)$$

where now δq describes a zero-mean Gaussian distribution with covariance C_q and δp describes a zero-mean Gaussian distribution with covariance C_p.

A Gaussian distribution with covariance C_p undergoing a linear transformation J_c will have covariance $J_c C_p J_c^T$. The Jacobian J_c is a function of the state, $J_c(x)$, as is $C_p(x)$, thus we can write $C_q(x)$ as a composite function of the state, completing the specification of the forward model $y = f(x)$.

For the numerical solution, we shall also require the Jacobian of the forward model, $J(x) = [\partial f(x)/\partial \delta x]$, which can be determined analytically and evaluated numerically when needed.

3.3 Initialization and estimation

For performing the inversion of this nonlinear system, we need an initial state value at which to start. From stereo, we can recover an excellent starting state using linear techniques.

The initial 3-D position is obtained via linear least squares from the 2-D means (the closest approach of the two image rays). The initial rotation is taken as a rotation about the z-axis only, in concordance with the image-plane rotation of one of the 2-D blob observations, say the left image. Finally, the initial shape is matched in the x–y plane to that of the left image (remember the initial 3-D blob is only rotated in z) and the z size is chosen to be the smaller of the two principal dimensions of the x–y plane.

Conceptually, then, we have back-projected the 2-D blob from the left image into a plane parallel to the left image at a depth determined by the correspondence of the 2-D blob means from the left and right images. We have then given the initial shape based on one projection and a heuristic notion of regularity.

The nonlinear estimation can now proceed in many ways. We have a forward model, $y = f(x)$, so we can develop a cost function based on the error $e(x, y) = y - f(x)$. We also have the Jacobian $J(x)$ analytically, so we have available to us many forms of nonlinear estimation techniques.

We use a form of the Levenberg-Marquardt (LM) algorithm for the static self-calibration problem and a form of the extended Kalman filter (EKF) for the dynamic problem of tracking motion and shape. We have also experimented with combining the iterative properties of LM with the recursive modeling of EKF to perform the two tasks together, but these results are still forthcoming.

4 Experimental analysis

4.1 3-D Tracking Performance

The motivating application for us has been understanding human behavior in uncontrived situations such as an office environment. We have developed a simple blob tracker based on the ideas presented in [9] that gives us reliable 2-D blobs of a person's face and hands in real-time. This tracker operating independently on two cameras in a wide-baseline configuration gives us correspondences that we can use to self-calibrate the stereo system and track people's movement and gestures at the same time. We present here some experimental results on 3-D tracking accuracy taken from our real-time system, which operates at 20–30 Hz using a single SGI O2 workstations.

4.1.1 Self-calibration

When a person first enters the space, the stereo calibration is obtained by collecting a set of three blob correspondences (face, left hand, right hand) over a number of frames (50–100 total correspondences) and computing the stereo calibration using the LM estimator on the batch of correspondences. Figure 1 shows a typical data set at system startup that calibrates the stereo rig.

The stereo pair shows the first image with overlayed blobs and large white boxes marking the current feature locations, and small white boxes representing the subsequent feature tracks.

The calibration points are collected in a time span of roughly 5 sec and the estimation requires less than 2 sec. In this case, the subject waved his arms up and down to generate data and the system quickly converged to the state shown in the bottom portion of Figure 1, which is a roughly overhead view showing the location of the cameras (COP and virtual image plane for each) and the 3-D trajectories of the hands and head.

4.1.2 Calibration Error Analysis

There is no absolute 3-D ground truth for self-calibration, but residual error can be analyzed in the image plane and relative error can be evaluated in 3-D.

The residual error approach consists of using the recovered camera parameters and blob locations to re-synthesize the feature locations and compare to the actual measurements. Over dozens of calibration sequences, the RMS image plane residual is 1.5

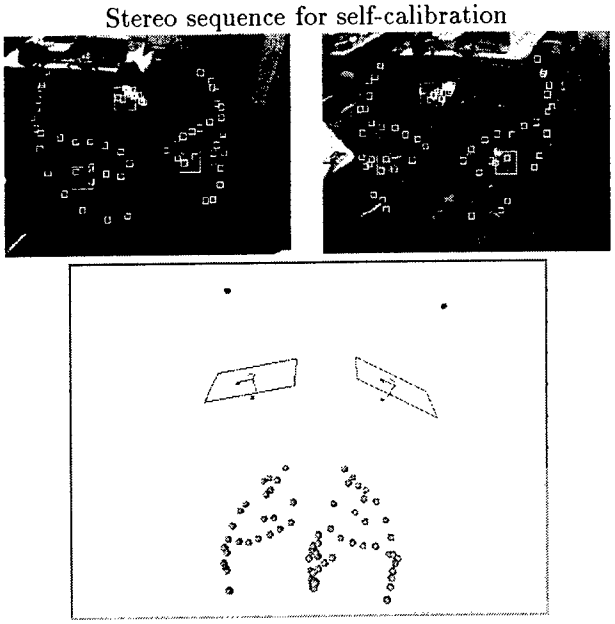


Figure 1: *The blob representation can be used to facilitate stereo self-calibration. Here we illustrate the self-calibration of a stereo rig in real time, simply from watching a person moving. The stereo pair shows the feature tracks on the person. The 3-D view shows a roughly overhead view of the space including the recovered cameras and the 3-D feature tracks. RMS residual error is 1.5 pixels; RMS 3-D errors are on the order of 2.25cm.*

pixels. The image size is (320,256) and the major blob axis diameters range from 20 to 35 pixels. The sources of residual errors include both measurement error (noise) and modeling error.

Relative 3-D error is evaluated after self-calibration by using the right hand as a 3-D pointer and traversing a trajectory of known shape and dimensions. In our case, a user moved his hand linearly along the edge of a table back and forth completing 4 cycles. The recovered 3-D trajectory is depicted in Figure 2, in which the three coordinates are plotted against frame number. The digitized hand locations are clustered along a 3-D line segment with known length (120cm). A line segment is fit to the cluster using 3-D regression and the coordinates are scaled to the known length.

The resulting RMS error from this analysis is 2.25cm. The relative error of reconstruction of hand position over this trajectory is therefore 1.8 percent. The sources of error include not only noise and modeling error, but also hand movement error, since the trajectory followed by the person was not exactly linear and the hand shape changes. Thus only a fraction of this relative error should be counted as computational error.

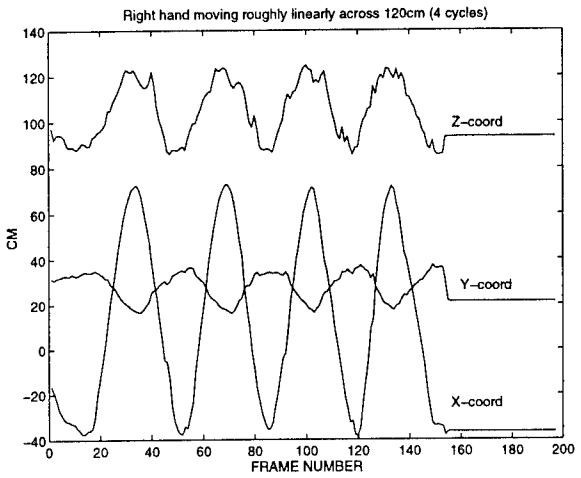


Figure 2: *Translation of the right hand back and forth along a linear trajectory after self-calibration. Reconstruction of hand position has an RMS error of 2.25cm, resulting in a relative error of 1.8% over this trajectory.*

4.1.3 3-D Person-tracking and shape recovery

The self-calibration can in principle take place simultaneously with shape and motion estimation, but for the purposes of performing our experiments they have been implemented separately. This makes it easier to isolate the sources of error for each component of the estimation.

Thus, in this section we are concerned with the steady-state characteristics of shape and motion estimation after self-calibration has converged. Again, since absolute motion parameters are impossible to know, we evaluate relative error by formulating some hand motions with known qualitative parameters.

4.1.4 3-D Tracking Error Analysis

The first test case consists of a linear motion where the user tries to keep his hand shape and orientation constant while sliding his hand along the straight edge of a rigid box. This is similar to the previous test case for self-calibration except that rotation and shape are measured.

Figure 3 shows a stereo pair and 3-D blob estimates for one frame of the sequence. Analysis on the translation was performed by fitting the 3-D location estimates to a line and computing the RMS error of translation, which was 1.5cm, resulting in a similar relative error metric to the self-calibration analysis. Analysis on the rotation and shape were performed by computing the mean values and RMS errors, which were about 2 degrees and 5% relative error respectively. Sources of error include measurement noise and modeling error. The comparatively high relative error in shape is probably due to the fact that the shape parameters are the least well-conditioned parameters in the state.

A second test case addresses the dynamic behav-

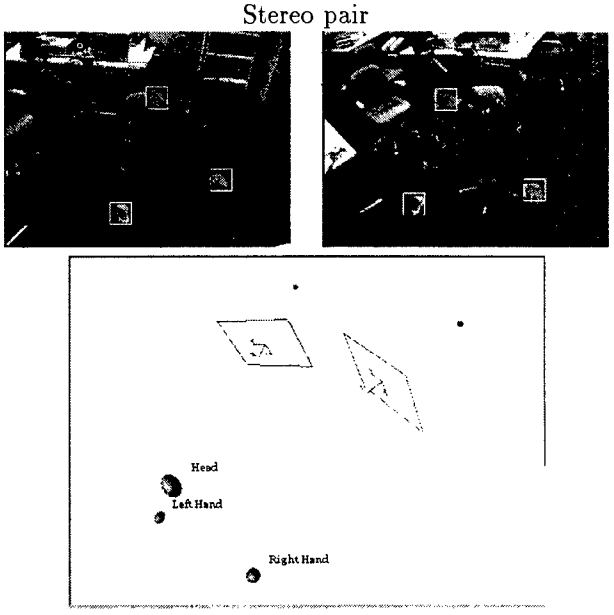


Figure 3: *Real-time estimation of position, orientation, and shape of moving human head and hands. We find RMS errors of 1.5cm, 2 degrees, and 5% on translation, rotation, and shape, respectively along a linear 3-D trajectory.*

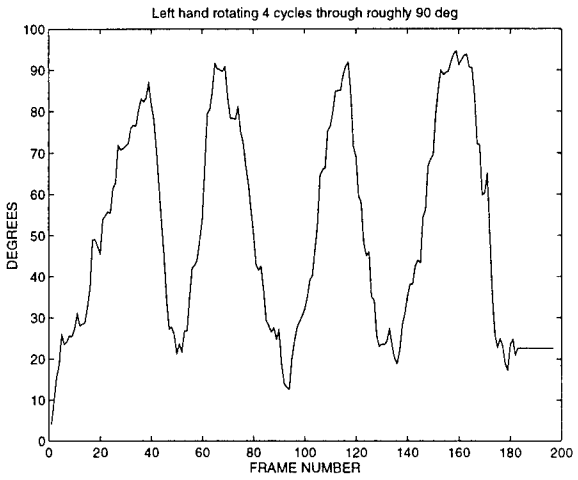


Figure 4: *Rotation of the left hand back and forth through roughly 90 deg. An analysis of the jitter in the angular signal results in measures of 2 deg RMS error, or roughly 2.2 percent relative error.*

ior of the rotation parameters. In this case, the user was told to rotate his hand from pointing directly forward to pointing roughly leftward and to cycle through this movement four times. The recovered angle of rotation is depicted in Figure 4, where a qualitative view of the estimation noise can be seen. Since the absolute ground truth is not known, the only way to quantitatively evaluate the error is to assume something about the trajectory.

In this case, we assume the actual trajectory was smooth, so we synthesize a set of error vectors by smoothing the trajectory and taking the difference between the actual and the smoothed trajectories. We use a 7-tap approximate Gaussian low-pass filter to smooth the trajectory. The RMS of the synthesized error is 1.98 deg, which results in a 2.2 percent relative error. The source of this error measure is primarily only the high-frequency components of measurement noise. Additional error may be present and may arise from low-frequency measurement errors or modeling errors.

4.2 Recognizing Gestures

We have used the recovered 3-D geometry for several different gesture recognition tasks, including a real-time person-independent American Sign Language reader [8], and a system that recognizes T'ai Chi gestures (and trains the user to perform them correctly!) [9].

Although we have been able to use standard HMM's to recognize such gestures with near-perfect accuracy, we have found the training of such models to be labor-intensive and difficult. This is because use of HMMs to describe multi-part signals (such as two-handed gestures) requires large amounts of training and even so the HMM parameter estimation process is typically unstable.

To improve on this situation, we have developed a new method of training a more general class of HMM, called the Coupled Hidden Markov Model. Coupled HMM's allow each hand to be described by a separate state model, and the interactions between them to be modeled explicitly and economically. The consequence is that much less training data is required, and the HMM parameter estimation process is much better conditioned. In Figure 4.1.4 shows recognizing T'ai Chi moves using the Coupled HMM method; for additional detail see Brand, Oliver, and Pentland 1997 [2].

5 Mouth Shape Extraction, Tracking, and Classification

In our system the mouth is modeled using the same blob methods that were used to find the head and hands, i.e. through a second-order models of the chromatic and spatial distribution. However for the mouth region it is critical that we employ a relatively sophisticated, multi-modal mixture model in order to achieve adequate performance.

We have developed thus a mixture-of-Gaussians shape/color model, that makes use of both positive and negative modeling. For the positive model we learn a description of the reddish lip region and the dark interior of the mouth; for the negative model we construct a description of the surrounding face area. We can then form a likelihood ratio that compares each pixel of the mouth region with both the positive and negative models, and achieve a good classification of the mouth shape. All of these models are learned on-the-fly using an incremental EM algorithm, allowing them to adapt to different skin types and different illumination conditions. Performance of this segmentation system is illustrated in

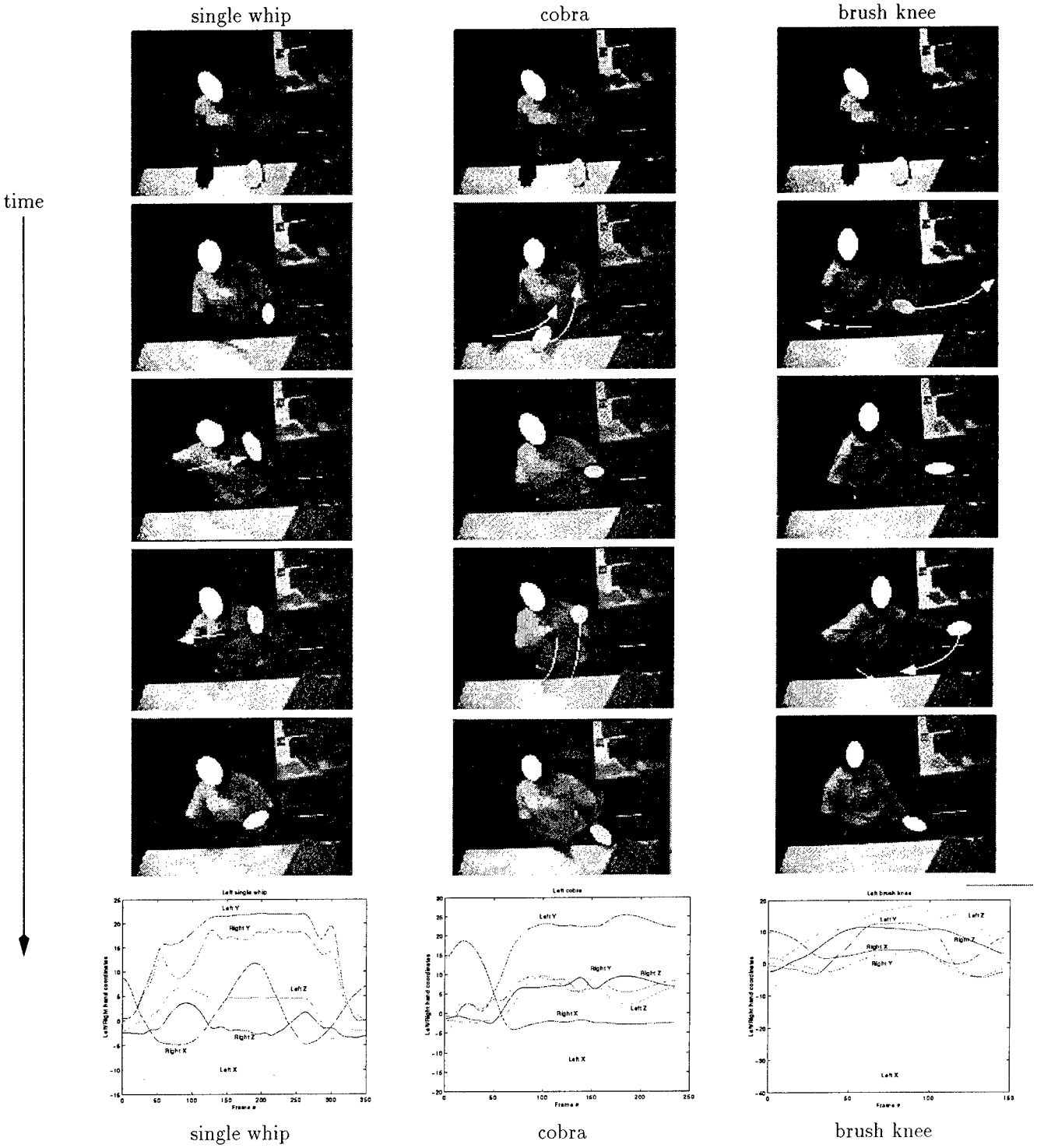


Figure 5: *Hand tracking of three gestures: Selected frames overlaid with hand blobs from vision. Graphs in the bottom row show the evolution of the feature vector over time. Sequences may be viewed at* `http://www.media.mit.edu/~brand/taichi.html`

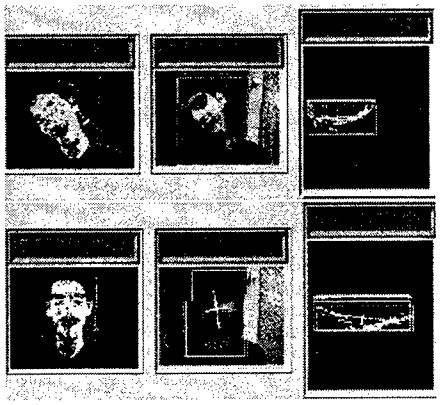


Figure 6: *Head and mouth tracking with rotations and facial hair*

Figure 6.

For recognition purposes the mouth shape is characterized by a feature vector consisting of the the area, the spatial eigenvalues of the mouth region and the x-y position of four extrema points. Rotation invariance is achieved by computing the face rotation angle and backprojecting the mouth region. Therefore the mouth always appears nearly horizontal during the recognition phase of processing, even though the user might turn their head.

5.1 Active Camera Control

For facial analysis it is neccessary to have a high-resolution view of the face. This view is provided by a third, active camera. The current estimation of the position and size of the user's face provides a reference signal to a PD controller which determines the tilt, pan and zoom of the camera so that the target (face) has the desired size and is at the desired location. Our system uses an abstraction of the camera parameters, in such a way that in the current version two different cameras (Canon VCC1 and Sony EVI-D30) can be successfully used in a totally transparent manner.

The zoom control is relatively simple, because it just has to be increased or decreased until the face reaches the desired size. Note however that the speed with which the camera zoom must be adjusted depends on the size of the target in the image. The relation between the zoom speed and the current camera zoom position follows a non-linear law which is approximated by a second-order polynomial function.

Pan and tilt speeds are controlled by:

$$V_c = \frac{C_e * F_s * E + C_d * \frac{dE}{dt}}{F_a} \qquad (12)$$

where C_e and C_d are camera-dependant constants, F_s is the system's running frequency, E is the the distance between the face current position and the face target position (e.g., the center of the image), F_z is the camera zoom factor, and S_c is the final speed transmitted to the camera.

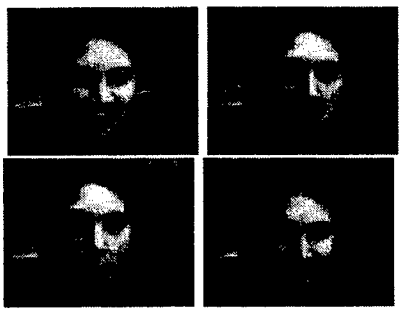


Figure 7: *Recognized mouth configurations: smile-open, sad, open and smile*

The system's running frequency is a key factor to be considered in order to stabilize the camera against frequency variations. Thus the control signals are low-pass filtered to ensure that they stay within Nyquist rate.

5.2 Expression Recognition

Our approach to temporal interpretation of facial expressions uses Hidden Markov Models (HMMs) to recognize different patterns of mouth movement. HMM's have been prominently and successfully used in speech recognition, which makes them quite appropriate to this task. We have developed a real-time HMM system that computes the maximum likelihood of the input sequence with respect to all the models during the testing or recognition phase. This HMM based system runs in real time on an SGI Indy, with the low-level vision processing occurring on a separate Indy, and communications occurring via a socket interface.

Using the mouth shape feature vector described above, we trained 5 different HMM's for each of the following mouth configurations (illustrated in figure 7): neutral or default mouth position, extended/smile mouth, sad mouth, open mouth and extended+open mouth (such as in laughing).

The neutral mouth acted to separate the various expressions, much as a silence model acts in speech recognition. The final HMM's we derived for the non-neutral mouth configurations consisted of 4-state forward HMM's. The neutral mouth was modeled by a 3-state forward HMM.

Recognition results for a eight different users making over 2000 expressions are summarized in table 1. The users were divided in different groups for training and testing purposes. The first of the recognition tasks shown in table 1 corresponds to a training and testing with all the eight users. The total number of examples is denoted by N, having a total N=2058 instances of the mouth expressions (N=750 for training and N=1308 for testing). The number of correctly recognized expressions is denoted by H.

6 Discussion

We have developed a real-time system for recovering 3-D object shape and motion and multiple-camera geometry from 2-D blob features. The 3-D objects and 2-D features are both represented using

Recognition Results	On training	On testing
All Users	%Correct=97.73 $[H = 733, N = 750]$	%Correct=95.95 $[H = 1255, N = 1308]$
Single Users	%Correct=100.00 $[H = 120, N = 120]$	%Correct=100.00 $[H = 240, N = 240]$

Table 1: *Recognition results on both training and testing data*

moment-based physical models called blobs. Nonlinear optimization techniques are used for estimation; the Levenberg-Marquardt technique is used for static parameters and the extended Kalman filter is used for dynamic estimation.

Experimental results verify that we can obtain good quantitative 3-D physical descriptions from these coarse 2-D image observations of people. We have experimentally demonstrated that this method can be used to self-calibrate stereo cameras from watching people move and subsequently to determine the location, orientation, and shape of parts of a person to an accuracy of a 2 cm, 2 degrees and a few percent, respectively (RMS errors).

These 3-D estimates have then been used as input to our HMM and Coupled HMM gesture recognition systems. Using this approach we have been able to obtain high accuracy at recognizing a wide variety of hand and face gestures, again in real-time using only modest computational resources.

Perhaps the most important performance evaluation, however, is that we have been able to build a person-tracking system using this technique which has run reliably for dozens of hours, with dozens of different subjects, in several different locations, and in real time (20–30 fps) using only standard workstations. The key to the robust, real-time performance is that the 2-D blob features on which the estimation relies can be reliably and efficiently extracted and matched in a bottom-up fashion.

We feel that use of this type of feature is a significant departure from the traditional notions of image features (e.g., points, lines) and image cues (e.g., motion fields, shading), and can lead to a basis for practical 3-D vision systems in application domains where traditional approaches have not had a great deal of success. Although the blob models provide only rigid motion and coarse shape information, they are fast and extremely reliable; thus futher precision and higher levels of details, if desired, can be safely bootstrapped from this level of representation, potentially leading to a powerful "coarse-to-fine" or "subsumption" approach to 3-D shape and motion analysis.

References

[1] Azerbayejani, A., and Pentland, A. (1996) "Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features," ICPR '96, Vienna, Austria.

[2] Brand, M., Oliver, N., and Pentland, A., (1997) "Coupled HMMs for Complex Action Recognition," to appear *CVPR '97*, San Juan, Puerto Rico.

[3] Bobick, A., and Bolles, R., (1992) "The Representation Space Paradigm of Concurrent Evolving Object Descriptions," *IEEE PAMI* 14(2):146-156.

[4] Kauth, R., Pentland, A., and Thomas, G., (1977) "BLOB: An Unsupervised Clustering Approach to Spatial Preprocessing of MSS Imagery," *11th Int'l Symp. on Remote Sensing of the Environment*, Ann Arbor, MI.

[5] Marr, D., (1982) "Vision," Freeman.

[6] Oliver, N., Bernard, F., Coutaz, J., and Pentland, A., (1997) "LAFTER: lips and face tracker," to appear *CVPR '97* San Juan, Puerto Rico.

[7] Pentland, A., (1976) "Classification by Clustering," *IEEE Proc. Symp. on Machine Processing of Remotely Sensed Data*, Purdue, Indiana.

[8] Starner, T., and Pentland, A., (1995) "Visual Recognition of American Sign Language Using Hidden Markov Models," *Proc. Int'l Workshop on Face and Gesture Recognition*, Zurich, Switzerland.

[9] Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A., (1995) "Pfinder: Real-Time Tracking of the Human Body," *Photonics East, SPIE Proc. Vol. 2615*, Bellingham, WA.

Modeling and Prediction of Human Behavior

Alex Pentland
Massachusetts Institute of Technology
Room E15-387, 20 Ames Street
Cambridge, MA 02139, USA
sandy@media.mit.edu

Andrew Liu
Nissan Cambridge Basic Research
4 Cambridge Center
Cambridge, MA, 02142
andy@pathfinder.cbr.com

Abstract

We describe our research toward building systems that include a complex, multi-state model of human dynamic behavior. This can allow us to predict human behavior over short periods of time, in order to create control systems that intelligently complement the human's action. To accomplish this requires inferring the internal state of the human, and then correctly adapting the remainder of the system to achieve optimal performance. We describe methods for achieving this goal, and report an initial experiment in which we were able to achieve 95% accuracy at predicting automobile driver's actions from their initial preparatory movements.

1 Introduction

Our approach is to modeling human behavior is to consider the human as a Markov device with a (possibly large) number of internal 'mental' states, each with its own particular control behavior, and inter-state transition probabilities (e.g., in a car the states might be passing, following, turning, etc.). A simple example of this type of human model would be a bank of standard quadratic controllers, each using different dynamics and measurements, together with a network of probabilistic transitions between them.

To integrate this human model into an optimal control system encopassing both human and machine it is necessary to know which controller is currently "in charge," and to predict transitions between controllers, so that the remainder of the system (e.g., a car) can configure itself to achieve its best overall performance. However the internal states of the human are not directly observable, so they must be determined through an indirect estimation process. One efficient and robust method of accomplishing this is to use the expectation-maximization methods developed for use of Hidden Markov Models (HMM) in speech processing.

By using these methods to identify a user's current internal (intentional) state, and to predict the most-likely subsequent internal state, we expect to be able to design systems that are able to dynamically reconfigure themselves to better fit the situation. This can potentially allow for higher performance than is possible with a fixed model of the human (assuming similar controller complexity).

2 Simple Dynamic Models

Among the simplest non-trivial models that have been considered for modeling human behavior are single dynamic processes

$$\mathbf{X}_{k+1} = \mathbf{f}(\mathbf{X}_k, \Delta t) + \xi(t) \tag{1}$$

where the function $\mathbf{f}$ models the dynamic evolution of state vector $\mathbf{X}_k$ at time k, and let us define an observation process

$$\mathbf{Y}_k = \mathbf{h}(\mathbf{X}_k, \Delta t) + \eta(t) \tag{2}$$

where the sensor observations $\mathbf{Y}$ are a function $\mathbf{h}$ of the state vector and time. Both ξ and η are white noise processes having known spectral density matrices.

Using Kalman's result, we can then obtain the optimal linear estimate $\hat{\mathbf{X}}_k$ of the state vector $\mathbf{X}_k$ by use of the following *Kalman filter*:

$$\hat{\mathbf{X}}_k = \mathbf{X}_k^* + \mathbf{K}_k(\mathbf{Y}_k - \mathbf{h}(\mathbf{X}_k^*, t)) \tag{3}$$

provided that the Kalman gain matrix $\mathbf{K}_k$ is chosen correctly [12]. At each time step k, the filter algorithm uses a state prediction $\mathbf{X}_k^*$, an error covariance matrix prediction $\mathbf{P}_k^*$, and a sensor measurement $\mathbf{Y}_k$ to determine an optimal linear state estimate $\hat{\mathbf{X}}_k$, error covariance matrix estimate $\hat{\mathbf{P}}_k$, and predictions $\mathbf{X}_{k+1}^*$, $\mathbf{P}_{k+1}^*$ for the next time step.

The prediction of the state vector $\mathbf{X}_{k+1}^*$ at the next time step is obtained by combining the optimal state estimate $\hat{\mathbf{X}}_k$ and Equation 1:

$$\mathbf{X}_{k+1}^* = \hat{\mathbf{X}}_k + \mathbf{f}(\hat{\mathbf{X}}_k, \Delta t)\Delta t \tag{4}$$

In our application this prediction equation is also used with larger times steps, to predict the human's future state. For instance, in a car such a prediction capability can allow us to maintain synchrony with the driver by giving us the lead time needed to alter suspension components, etc.

Finally, given the state vector $\mathbf{X}_k$ at time k we can predict the measurements at time $k + \Delta t$ by

$$\mathbf{Y}_{k+\Delta t} = \mathbf{h}(\mathbf{X}_k, \Delta t) \tag{5}$$

and the predicted state vector at time $k+\Delta t$ is given by

$$\hat{\mathbf{X}}_{k+\Delta t} = \mathbf{X}_k^* + \mathbf{f}(\hat{\mathbf{X}}_k, \Delta t)\Delta t \tag{6}$$

3 Multiple Dynamic Models

Human behavior, in all but the simplest tasks, is not as simple as a single dynamic model. The next most complex model of human behavior is to have *several* alternative models of the person's dynamics, one for each class of response. Then at each instant we can make observations of the person's state, decide which model applies, and then make our response based on that model. This is known as the *multiple model* or *generalized likelihood* approach, and produces a generalized maximum likelihood estimate of the current and future values of the state variables [13]. Moreover, the cost of the Kalman filter calculations is sufficiently small to make the approach quite practical.

Intuitively, this solution breaks the person's overall behavior down into several "prototypical" behaviors. For instance, in the driving situation we might have dynamic models corresponding to a relaxed driver, a very "tight" driver, and so forth. We then classify the driver's behavior by determining which model best fits the driver's observed behavior.

Mathematically, this is accomplished by setting up a set of states S, each associated with a Kalman filter and a particular dynamic model:

$$\hat{\mathbf{X}}_k^{(i)} = \mathbf{X}_k^{*(i)} + \mathbf{K}_k^{(i)}(\mathbf{Y}_k - \mathbf{h}^{(i)}(\mathbf{X}_k^{*(i)}, t)) \quad (7)$$

where the superscript (i) denotes the i^{th} Kalman filter. The *measurement innovations process* for the i^{th} model (and associated Kalman filter) is then

$$\Gamma_k^{(i)} = \mathbf{Y}_k - \mathbf{h}^{(i)}(\mathbf{X}_k^{*(i)}, t) \quad (8)$$

The measurement innovations process is zero-mean with covariance $\mathcal{R}$.

The i^{th} measurement innovations process is, intuitively, the part of the observation data that is unexplained by the i^{th} model. The model that explains the largest portion of the observations is, of course, the model most likely to be correct. Thus, at each time step, we calculate the probability $Pr^{(i)}$ of the m-dimensional observations $\mathbf{Y}_k$ given the i^{th} model's dynamics,

$$Pr^{(i)}(\mathbf{Y}_k) = \frac{\exp\left(-\frac{1}{2}\Gamma_k^{(i)T}\mathcal{R}^{-1}\Gamma_k^{(i)}\right)}{(2\pi)^{m/2}\mathrm{Det}(\mathcal{R})^{1/2}} \quad (9)$$

and choose the model with the largest probability. This model is then used to estimate the current value of the state variables, to predict their future values, and to choose among alternative responses.

Note that when optimizing predictions of measurements Δt in the future, Equation 8 must be modified slightly to test the predictive accuracy of state estimates from Δt in the past.

$$\Gamma_k^{(i)} = \mathbf{Y}_k - \mathbf{h}^{(i)}(\mathbf{X}_{k-\Delta t}^{*(i)} + \mathbf{f}^{(i)}(\hat{\mathbf{X}}_{k-\Delta t}^{(i)}, \Delta t)\Delta t, t)) \quad (10)$$

by substituting Equation 6.

3.1 Results Using Multiple Dynamic Models

We have used this method to accurately remove lag in a high-speed telemanipulation task by continuously re-estimating the user's arm dynamics (e.g., tense and stiff, versus relaxed and inertia-dominated) [3].

In this case, the state vector $\mathbf{X}_k$ consists of the true position, velocity, and acceleration of the hand in each of the x, y, and z coordinates, and the observation vector $\mathbf{Y}_k$ consists of the position readings for the x, y, and z coordinates. We found that using this multiple-model approach we were able to obtain significantly better predictions of the user's hand position that was possible using a single dynamic or static model.

4 Hidden Markov Dynamic Models

In the above multiple dynamic model, all the processes have a fixed likelihood at each time step. However, this is uncharacteristic of most situations, where there is a fixed sequence of internal states each with its own dynamics. Consider driving through a curve; the driver may be modeled as having transitioned through a series of states $\lambda = (s_1, s_2, ...s_k), s_i \epsilon S$, for instance, entering a curve, in the curve, and exiting a curve, and other. Transitions between these states happened only in the order indicated, with a final transition from other to entering the curve.

Thus in considering state transitions among a set of dynamic models we should make use of our current estimate of the driver's internal state. We can accomplish this fairly generally by considering the Markov probability structure of the transitions between the different states. The input to decide the person's current internal state (e.g., which dynamic model currently applies) will be the measurement innovations process as above, but instead of using this directly in Equation 9 we will instead also consider the Markov inter-state transition probabilities.

While a substantial body of literature exists on HMM technology [5, 6, 8, 11], we will first briefly outline a traditional discussion of the algorithms. After outlining the fundamental theory in training and testing of a discrete HMM, we will generalize these results to the continuous density case applicable to switching between dynamic models. For broader discussion of the topic, [6, 9] are recommended.

A time domain process demonstrates a Markov property if the conditional probability density of the current event, given all present and past events, depends only on the j^{th} most recent events. If the current event depends solely on the most recent past event, then the process is a first order Markov process.

The initial topology for an HMM can be determined by estimating how many different states are involved in the observed phenominon. Fine tuning this topology can be performed empirically. Figure 1, for instance, shows a four state HMM with skip transitions that we have used to classify complex hand motions.

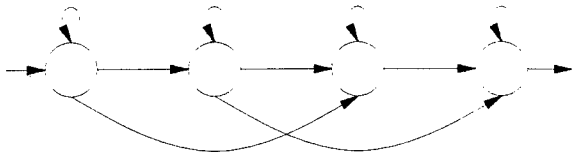


Figure 1: The four state HMM used for recognition, from [4].

There are three key problems in HMM use. These are the evaluation, estimation, and the decoding problems. The evaluation problem is that given an observation sequence and a model, what is the probability that the observed sequence was generated by the model ($Pr(\mathbf{Y}|\lambda)$) (notational style adapted from [6])? If this can be evaluated for all competing models for an observation sequence, then the model with the highest probability can be chosen for recognition.

The Viterbi algorithm provides a quick means of evaluating a set of HMM's in practice as well as providing a solution for the decoding problem. In decoding, the goal is to recover the state sequence given an observation sequence. The Viterbi algorithm can be viewed as a special form of the forward-backward algorithm where only the maximum path at each time step is taken instead of all paths. This optimization reduces computational load and additionally allows the recovery of the most likely state sequence. The steps to the Viterbi are

- Initialization. For all states i, $\delta_1(i) = \pi_i b_i(Y_1)$; $\psi_i(i) = 0$
- Recursion. From $t = 2$ to k and for all states j, $\delta_t(j) = \max_i[\delta_{t-1}(i)a_{ij}]b_j(Y_t)$; $\psi_t(j) = \text{argmax}_i[\delta_{t-1}(i)a_{ij}]$
- Termination. $P = \max_{s \epsilon S}[\delta_k(s)]$; $s_k = \text{argmax}_{s \epsilon S}[\delta_k(s)]$
- Recovering the state sequence. From $t = k - 1$ to 1, $s_t = \psi_{t+1}(s_{t+1})$

Note that since Viterbi only guarantees the maximum of $Pr(\mathbf{Y}, S|\lambda)$ over all state sequences S (as a result of the first order Markov assumption) instead of the *sum* over all possible state sequences, the resultant scores are only an approximation. However, [8] shows that this is often sufficient.

4.1 The Continuous Case

So far this discussion of HMMs has assumed some sort of quantization of feature vectors into classes, wheras the innovations processes that wil drive our inter-state transitions are continuous. Consequently, instead of using vector quantization, we must employ the actual probability densities for the innovations processes. Fortunately, Baum-Welch parameter estimation, the Viterbi algorithm, and the forward-backward algorithms can be modified to handle a variety of characteristic densities [7]. In this paper, however, the densities will be assumed to be Gaussian. Specifically, from Equation 9,

$$b_j(Y_t) = \frac{\exp\left(-\frac{1}{2}\Gamma_k^{(i)T}\mathcal{R}^{-1}\Gamma_k^{(i)}\right)}{(2\pi)^{m/2}\text{Det}(\mathcal{R})^{1/2}} \qquad (11)$$

Initial estimations of μ and σ may be calculated by dividing the evidence evenly among the states of the model and calculating the mean and variance in the normal way. Whereas flat densities were used for the initialization step before, the evidence is used here. Now all that is needed is a way to provide new estimates for the output probability. This can be accomplished by the Kalman filter update equations.

5 An Experiment Using Hidden Markov Dynamic Models

We are now using this approach to identify automobile driver's current internal (intentional) state, and their most-likey subsequent internal state. In the case of driving the macroscopic actions are events like turning left, stopping, or changing lanes. The internal states are the individual steps that make up the action, and the observed behaviors will be changes in heading and acceleration of the car.

The intuition is that even apparently simple driving actions can be broken down into a long chain of simpler subactions. A lane change, for instance, may consist of the following steps (1) a preparatory centering the car in the current lane, (2) looking around to make sure the adjacent lane is clear, (3) steering to initiate the lane change, (4) the change itself, (5) steering to terminate the lane change, and (6) a final recentering of the car in the new lane. In our current study we are statistically characterizing the sequence of steps within each action, and then using the first few preparatory steps to identify which action is being initiated.

To recognize which action is occuring one compares the observed pattern of driver behavior to hidden Markov dynamic models of each action, in order to determine which action is *most likely* given the observed pattern of steering and acceleration/braking. This matching can be done in real-time on current microprocessors, thus potentially allowing us to recognize a drivers' intended action from their preparatory movements.

If the pattern of steering and acceleration is monitored internally by the automobile, then the ability to recognize which action the driver is beginning to initiate can allow intelligent cooperation by the vehicle. If heading and acceleration is monitored externally via video cameras, as in Figures 2 and 3 (the 'blob' processing algorithm that extracts the vehicle parameters from video is described in Boer, Fernandez, Pentland, and Liu 1996 [2]), then we can more intelligently control the traffic flow.

5.1 Experimental Design

The goal is to test the ability of our framework to characterize driver's steering and acceleration/braking patterns in order to classify the driver's intended action. The experiment was conducted within the Nissan Cambridge Basic Research driving simulator, shown in Figure 4(a). The simulator

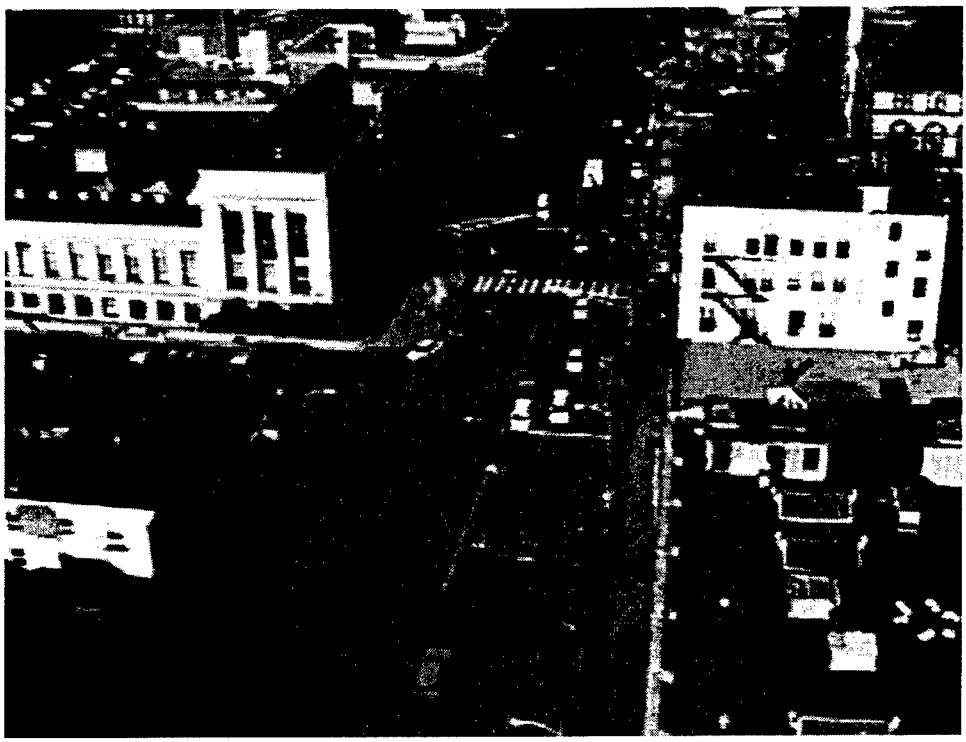


Figure 2: Video data of Central Square, Cambridge MA.

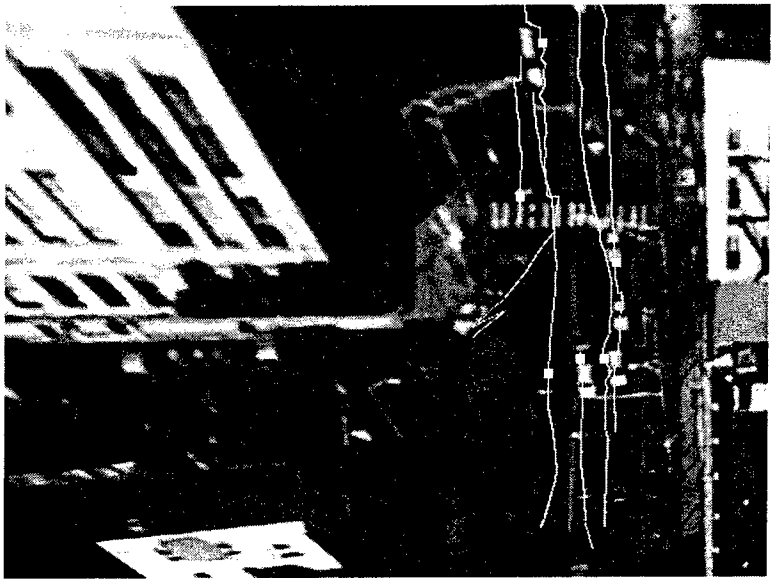


Figure 3: Vehicle tracks extracted from spatially-rectified video data.

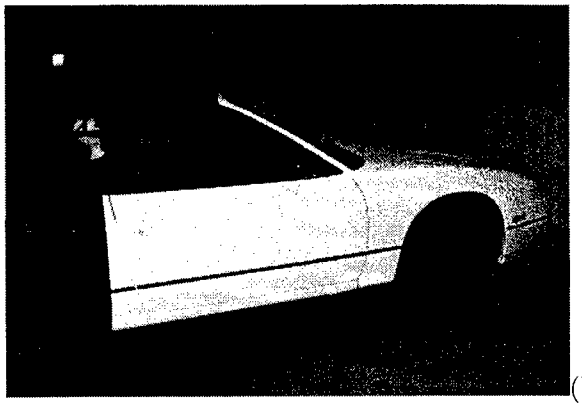
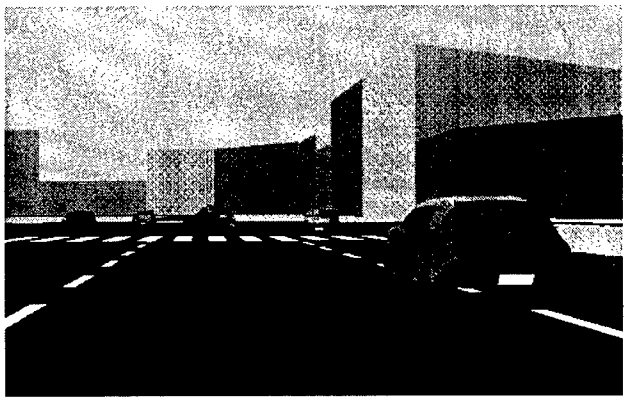
(a)  (b) 

Figure 4: (a) Nissan Cambridge Basic Research simulator, (b) part of the simulated world seen by the subjects.

consists of the front half of a Nissan 240SX convertible and a 60 deg (horizontal) by 40 deg (vertical) image projected onto the wall facing the driver. The 240SX is instrumented to record driver control input such as steering wheel angle, brake position, and accelerator position.

Eight adult male subjects were instructed to use this simulator to drive through an extensive computer graphics world, illustrated in Figure 4(b). This world contains a large number of buildings, many roads with standard markings, and other moving cars. Each subject drove through the simulated world for approximately 20 minutes, during which time the driver's control of steering angle and steering velocity, car velocity and car acceleration were recorded at $1/10^{th}$ second intervals.

From time to time during this drive text commands were presented on-screen, whereupon the subjects had to assess the surrounding situation, formulate a plan to carry out the command, and then act to execute the command. The variables of command location, road type, surrounding buildings, and traffic conditions were varied randomly throughout the experiment.

The commands included: (1) stop at next intersection, (2) turn left at next intersection, (3) turn right at next intersection, (4) change lanes, (5) pass the car in front of you, and (6) drive normally with no turns or lane changes. A total of 72 stop, 262 turn, 47 lane change, 24 passing, and 208 drive-normal episodes were recorded. The time needed to complete each command varied from approximately 5 to 10 seconds, depending upon the complexity of both the action and the surrounding situation.

Using the steering and acceleration data recorded while subjects carried out these commands, we built three-state models of each type of driver action (stopping, turn left, turn right, lane change, car passing, and drive-normal) using the estimation tools provided by the Entropic's HTK computer software [11].

To assess the classification accuracy of these models we combined them with the Viterbi recognition algorithm, and examined the stream of drivers' steering and acceleration innovations in order to detect and classify the driver's actions. We then examined the computer's classifications immediately after each command, and recorded whether or not the computer had correctly labeled the action.

Recognition results were tabulated at one second after the presentation of a command to the subject. Note that the minimum response time to a command is approximately 0.5 seconds, so that the one-second point is at most one-half second after the beginning of the driver's action. The one-second point, therefore, is roughly $1/10^{th}$ of the way through the action.

To obtain unbiased estimates of recognition performance, we employed the "leaving one out" method. In this method models are trained on seven subjects and then tested on the eighth subject. This is then repeated eight times, each time leaving out a different one of the eight subjects, and the eight sets of recognition statistics are averaged.

5.2 Results

At one second after the command presentation ($\approx$ 0.5 seconds after the beginning of action, and roughly 10% of the way through the action) mean recognition accuracy was $95.24\% \pm 3.1\%$. These results demonstrate that many types of driving behavior are sufficiently stereotyped that they are reliably recognizable from observation of the driver's preparatory movements.

To test whether our sample is sufficiently large to adequately encompass the range of between-driver variation, we compared these results to the case in which we train on all subjects and then test on the training data. In the test-on-training case the recognition accuracy was 98.8%, indicating that we have a sufficiently large sample of driving behavior in this experiment.

While these results are very promising, caution must be taken in transferring them to real-world driving. It is possible, for instance, that there are driving styles not seen in any of our subjects. Similarly, the driving conditions found in our simulator do not span the entire range of real driving situations. We belive, however, that our simulator is sufficiently realistic that compariable accuracies can be obtained in real driving. Moreover, there is no

strong need for models that suit all drivers; most cars are driven by a relatively small number of drivers, and this fact can be used to increase classification accuracy.

6 Conclusion

We have demonstrated that we can accurately categorize drivers' actions very soon after the beginning of the action using our behavior modeling methodology. Because of the generic nature of the driving task, there is reason to believe that this approach to modeling human behavior will generalize to a wide variety of human-machine systems. This would allow us to automatically recognize the people's *intended action*, and thus to build control systems that dynamically adapt to better suit the human's purpose.

Acknowledgements. We would particularly like to thank Thad Starner for his work in developing HMM technology for gesture understanding.

References

[1] M. Land. Predictable eye-head coordination during driving, *Nature*, Vol. 359, pp. 318-320, 1992.

[2] Boer, E., Fernandez, M., Pentland, A., and Liu, A., (1996) "Method for Evaluating Human and Simulated Drivers in Real Traffic Situations," *IEEE Vehicular Tech. Conf.*, Atlanta, GA.

[3] M. Friedmann, T. Starner, and A. Pentland. Device Synchronization using an Optimal Linear Filter, *Proc. ACM 1992 Symposium on Interactive 3D Graphics*, Boston, MA, May 1992.

[4] T. Starner, and A. Pentland. Visual Recognition of American Sign Language Using Hidden Markov Models, *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, Zurich, Switzerland, June 26-28, 1995.

[5] L. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of markov processes. *Inequalities*, 3:1–8, 1972.

[6] X. Huang, Y. Ariki, and M. Jack. *Hidden Markov Models for Speech Recognition.* Edinburgh Univ. Press, Edinburgh, 1990.

[7] B. Juang. Maximum likelihood estimation for mixture multivariate observations of markov chains. *AT&T Technical Journal*, 64:1235–1249, 1985.

[8] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, p. 4–16, Jan. 1996.

[9] T. Starner. Visual Recognition of American Sign Language Using Hidden Markov Models. Master's thesis, MIT Media Laboratory, Feb. 1995.

[10] T. Starner, J. Makhoul, R. Schwartz, and G. Chou. On-line cursive handwriting recognition using speech recognition methods. In *ICASSP*, 1994.

[11] S. Young. *HTK: Hidden Markov Model Toolkit V1.5.* Cambridge Univ. Eng. Dept. Speech Group and Entropic Research Lab. Inc., Washington DC, Dec. 1993.

[12] R.E. Kalman and R.S. Bucy. New results in linear filtering and prediction theory. In *Transaction ASME (Journal of basic engineering)*, 83D, 95-108, 1961.

[13] A.S. Willsky. Detection of Abrupt Changes in Dynamic Systems. In M. Basseville and A. Benveniste, (Ed.). *Detection of Abrupt Changes in Signals and Dynamical Systems*, Lecture Notes in Control and Information Sciences, No. 77, pp. 27-49, Springer-Verlag, 1986.

A Trainable System for People Detection

Michael Oren Constantine Papageorgiou Pawan Sinha
Edgar Osuna Tomaso Poggio

CBCL and AI Lab
MIT
Cambridge, MA 02139

Abstract

This paper presents a trainable object detection architecture that is applied to detecting people in static images of cluttered scenes. This problem poses several challenges. People are highly non-rigid objects with a high degree of variability in size, shape, color, and texture. Unlike previous approaches, this system learns from examples and does not rely on any a priori (hand-crafted) models or on motion.

The detection technique is based on a wavelet representation of the image: by learning an object class in terms of a subset of an overcomplete dictionary of wavelet basis functions. It is invariant to changes in color and texture and can be used to robustly define a rich and complex class of objects such as people. We show how the invariant properties and computational efficiency of the wavelet representation make it an effective tool for object detection.

1 Introduction

The problem of object detection has seen a high degree of interest over the years. The fundamental problem is how to characterize an object class. In contrast to the case of pattern classification, where we need to decide between a relatively small number of classes, the detection problem requires us to differentiate between the object class and the rest of the world. As a result, the class description for object detection must have large discriminative power to handle the cluttered scenes it will be presented with. Furthermore, in modeling complicated classes of objects (e.g. faces, pedestrians) the intra-class variability itself is significant and difficult to model. Since it is not known how many instances of the class are presented in the scene, if any, the detection problem cannot easily be solved using methods such as maximum-a-posteriori probability (MAP) or maximum likelihood models. Consequently, the classification of each pattern in the image must be done independently; this makes the decision problem susceptible to missed instances of the class and false positives.

There has been a body of work on people detection (Tsukiyama & Shirai, 1985[Tsukiyama and Shirai-1985], Leung & Yang, 1987[Leung and Yang-1987b][Leung and Yang-1987a], Rohr, 1993[Rohr-1993], Chen & Shirai, 1994[Chen and Shirai-1994]); these approaches are heavily based on motion and hand crafted models. An important aspect of our system is that the model is automatically learned from examples and avoids the use of motion and explicit segmentation.

One of the successful systems in the area of trainable object detection in cluttered scenes is the face detection system of Sung and Poggio [Sung and Poggio-1994]. They model face and non-face patterns in a high dimensional space and derive a statistical model for the class of frontal human faces. Similar face detection systems have been developed by others (Vaillant, et al.[Vaillant *et al.*-1994], Rowley, et al.[Rowley *et al.*-1995], Moghaddam and A. Pentland[Moghaddam and Pentland-1995], Osuna et al.[Edgar Osuna and Girosi-1996]).

Frontal human faces, despite their variability, share very similar patterns (shape and the spatial layout of facial features) and their color space is very constrained. This is not the case with pedestrians. Figure 1 shows several typical images of people in our database. These images illustrate the difficulties of pedestrian detection; there is significant variability in the patterns and colors within the boundaries of the body. The detection problem is also complicated by the absence of constraints on the image background. Given these problems, direct analysis of pixel characteristics (e.g. intensity, color and texture) is not adequate. This paper presents a new approach based on learning a class-specific wavelet representation. This representation is motivated by an earlier piece of work by one of the authors [Sinha-1994a] [Sinha-1994b] who derived a new invariant called the 'ratio template' and applied it to face detection.

A ratio template encodes the ordinal structure of the brightness distribution on a face. It consists of a set of inequality relationships between the average intensities of a few different face-regions. This design was motivated by the observation that while the absolute intensity values of different regions change dramatically under varying illumination conditions, their mutual ordinal relationships (binarized ratios) remain largely unaffected. Thus, for instance, the forehead is typically brighter than the eye-socket regions for all but the most contrived lighting setups. A small set of such relationships, collectively called a ratio template, provides a powerful constraint for face detection. The emphasis

[0]This research was sponsored by DARPA and ONR.

Figure 1: Examples of images of people in the training database. The examples vary in color, texture, view point (either frontal or rear) and background.

on the use of qualitative relationships also renders the ratio template construct perceptually plausible (the human visual system is poor at judging absolute brightnesses but remarkably adept at making ordinal brightness comparisons).

The success of the template-ratio approach for face detection and the shortcoming of the standard pixel-based image representation suggest the use of basis functions that encode differences in the average intensities between neighboring regions. The Haar wavelet is a particular simple family of such basis functions that we choose for our system. The Haar wavelet representation has also been used for image database retrieval, Jacobs *et al.*[Jacobs *et al.*-1995], where the largest wavelet coefficients were used as a measure of similarity between two images. In our work, we use the wavelet representation to capture the structural similarities between various instances of the class. Another important feature of our work is the use of an overcomplete, or redundant, set of basis functions which is important in capturing global constraints on the object shape and provides adequate spatial resolution. Our results on pedestrian detection using the wavelet representation demonstrate that it may be a promising framework for computer vision applications.

2 Wavelets

In this section, we review the Haar wavelet, describe a denser (redundant) transform, and describe the wavelet representation.

2.1 The Haar Wavelets

We provide only a concise description of wavelets; a more detailed treatment can be found in [Mallat-1989]. The definition of wavelets is closely related to the concept of multi-resolution analysis that is based on a sequence of approximating subspaces, $\ldots V_j \subset V_{j+1} \subset \ldots$, such that each subspace, V_j, describes finer details than the preceding one. The wavelet subspaces, $\{W_j\}_j$, are defined to be the orthogonal complement of two consecutive approximating subspaces, $V_{j+1} = V_j \bigoplus W_j$, and can be interpreted as the subspace of "details" in increasing refinements. Each approximating subspace, V_j, is spanned by a basis of *scaling functions*, $\{\phi(x)_{j,l}\}_l$, and, similarly, each wavelet subspace, W_j, is spanned by a basis of *wavelet functions*, $\{\psi(x)_{j,l}\}_l$. The union of the wavelet functions comprises a basis for $L_2(R)$. It can be shown (under the standard condition of multi-resolution analysis) that all the scaling functions can be generated from dilations and translations of one scaling function. Similarly, all the wavelet functions are dilations and translations of the mother wavelet function. The structure of the approximating and wavelet subspaces leads to an efficient cascade algorithm for the computation of the scaling coefficients, $\lambda_{j,k}$, and the wavelet coefficients, $\gamma_{j,k}$:

$$\lambda_{j,k} = \sum_{n \in Z} h_{n-2k} \lambda_{j+1,n} \quad (1)$$

$$\gamma_{j,k} = \sum_{n \in Z} g_{n-2k} \lambda_{j+1,n} \quad (2)$$

where $\{h_i\}$ and $\{g_i\}$ are the filter coefficients corresponding to the scaling and wavelet functions. It is important to observe that the discrete wavelet transform (DWT) performs *downsampling* or *decimation* of the coefficients at the finer scales since the filters h and g are moved in a step size of 2 for each increment of k.

In this paper we use the Haar wavelets; the corresponding filters are: $h = \{\ldots, 0, \frac{1}{2}, \frac{1}{2}, 0, 0, \ldots\}$ and $g = \{\ldots, 0, -\frac{1}{2}, \frac{1}{2}, 0, 0, \ldots\}$

The scaling coefficients are simply the averages of pairs of adjacent coefficients in the coarser level while the wavelet coefficients are the differences.

2.1.1 2-Dimensional Wavelet Transform

The natural extension of wavelets to 2-dimensional signals is obtained by taking the tensor product of two 1-dimensional wavelet transforms. The result is the three types of wavelet basis functions shown in Figure 2: the tensor product of a wavelet by a scaling function, $\psi(x,y) = \psi(x) \otimes \phi(y)$, is a *vertical* coefficient, a scaling function by a wavelet, $\psi(x,y) = \phi(x) \otimes \psi(y)$, is a *horizontal* coefficient, and a wavelet by a wavelet, $\psi(x,y) = \psi(x) \otimes \psi(y)$, is a *corner* coefficient.

Since the wavelets that the standard transform generates have irregular support, we use the non-standard 2-dimensional DWT where, at a given scale, the transform is applied to each dimension sequentially before proceeding to the next scale [Stollnitz *et al.*-1994]. The results are Haar wavelets with square support at all scales.

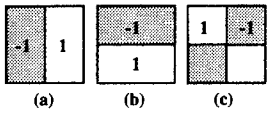


Figure 2: The 3 types of 2-dimensional non-standard Haar wavelets; (a) "vertical", (b) "horizontal", (c) "corner".

2.1.2 A Wavelet Dictionary

The standard Haar basis is not dense enough for our application. For the 1-dimensional transform, the distance between two neighboring wavelets at level n (with support of size 2^n) is 2^n. For better spatial resolution, we need a set of redundant basis functions, or an overcomplete *dictionary*, where the distance between the wavelets at scale n is $\frac{1}{4}2^n$, see Figure 3. We call this a *quadruple density* dictionary. As one can easily observe, the straightforward approach of shifting the signal and recomputing the DWT will *not* generate the desired dense sampling. Instead, this can be obtained by modifying the DWT. To generate wavelets with *double density*, where wavelets of level n are centered every $\frac{1}{2}2^n$, we simply do not downsample in equation 2. To generate the quadruple density dictionary, we do not downsample in equation 1 and get double density scaling coefficients. The next step is to calculate double density wavelet coefficients on the two sets of scaling coefficients — even and odd — separately. By interleaving the results of the two transforms we get quadruple density wavelet coefficients. For the next scale we keep only the even scaling coefficients of the previous level and repeat the quadruple transform on this set only; the odd scaling coefficients are dropped off. Since only the even coefficients are carried along at all the scales, we avoid an "explosion" in the number of coefficients, yet provide a dense and uniform sampling of the wavelet coefficients at all the scales. As with the regular DWT, the time complexity is $O(n)$ in the number of pixels n. The extension for the 2-dimensional transform is straightforward.

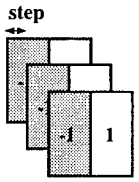


Figure 3: Quadruple density 2D Haar basis.

2.2 The Wavelet Representation

The Haar coefficients preserve all the information in the original pixel-based representation but they encode it as the difference in the average intensity between two neighboring regions and at different scales. For the description of an object class we can impose or learn various constraints on the value of the wavelet coefficients. The constraints can be very specific, for examples, "the value of the wavelet coefficient must lay in the range ...", or can be more qualitative, such as, "the coefficient must be different from zero." If we compute the wavelet transform on the log of the image intensities, the wavelet coefficients encode the ratios between the intensities instead the differences. The use of an overcomplete basis allows us to propagate constraints between neighboring regions and to describe complex patterns. As a result, the use of difference or ratio coding of intensities in different scales provides a very flexible and expressive representation that can characterize complex object classes. Furthermore, the wavelet representation is computationally efficient for the task of object detection since we do not need to compute the transform for different image regions but only once for the whole image and look at different sets of coefficients for different spatial locations. We choose the quadruple density wavelet transform since it is found to provide adequate spatial resolution.

2.2.1 Learning the Pedestrian Class Representation

Given an object class, the central problem is how to learn which are the relevant coefficients that express structure common to the entire object class. In this section, we describe the learning of the pedestrian class. Currently, we divided it into a two-stage learning process: identifying the wavelet coefficients and learning the relationships.

Since the images our system analyzes are of pedestrians in arbitrary cluttered scenes in unconstrained environments, it is easy to see that there are no consistent patterns in the color and texture of pedestrian bodies or the backgrounds against which they stand. This lack of clearly discernible interior features is circumvented by relying on (1) differences in the intensity between pedestrian bodies and their backgrounds and (2) consistencies within regions inside the body boundaries. Since the precise values of the wavelet coefficients and their signs have little meaning in this problem, we interpret the coefficients as either indicating an almost uniform area, i.e. "no-change", if their absolute value is relatively small or as indicating "strong change" if their absolute value is relatively large. The wavelet template we seek to identify will consist solely of wavelet coefficients (either vertical, horizontal or corner) whose types ("change"/"no-change") are clearly identified and are *consistent* along the ensemble of pedestrian images. Coefficients that are not consistent or important are not used. The identification of the wavelet template consists of addressing the following three major questions:

- The pedestrian images we use are of different subjects taken under different conditions; how

can we quantify the relative value of a coefficient as indicating "change" or "no-change" and how can we measure its consistency across the ensemble of examples?

- Can we find a set of constraints on the wavelet coefficients, or wavelet representation, that characterizes the pedestrian class?
- If such a wavelet representation is found, is its discriminative power high enough to detect pedestrians in cluttered scenes?

In the remainder of this section, we address the first question of identifying the important basis functions. In the next section, we show how a classifier can learn the constraints on the wavelet coefficients. In the section on the experimental results, we show that the wavelet representation is an effective and discriminative representation for the class.

The basic analysis to identify the important coefficients consists of two steps: first, we normalize the wavelet coefficients relative to the rest of the coefficients in the patterns; second, we analyze the averages of the normalized coefficients along the ensemble. We have collected a set of 564 color images of people (Figure 1) for use in the learning. All the images are scaled and clipped to the dimensions 128×64 such that the people are centered and approximately the same size (the distance from the shoulders to feet is about 80 pixels). In our analysis, we restrict ourselves to the wavelets at scales of 32×32 pixels (one array of 15×5 coefficients for each wavelet class) and 16×16 pixels (29×13 for each class). For each color channel (RGB) of every image, we compute the quadruple dense Haar transform and take the coefficient value to be the largest absolute value among the three channels. The normalization step involves computing the average of each coefficient's class ($\{vertical, horizontal, corner\} \times \{16, 32\}$) over all the pedestrian patterns and dividing every coefficient by its corresponding class average. We calculate the averages separately for each class since the power distribution between the different classes may vary.

To begin specifying the wavelet representation, we calculate the average of each normalized coefficient over the set of pedestrians. A base set of 597 color images of natural scenes of size 128×64 that do not contain people were gathered to compare with the pedestrian patterns. These non-pedestrian patterns are processed in the manner detailed above. Tables 1(a) and 1(b) show the average coefficient values for the set of vertical Haar coefficients of scale 32×32 for both the non-pedestrian and pedestrian classes. Several conclusions can be drawn from the tables. Table 1(a) shows that the process of averaging the coefficients within the pattern and then in the ensemble does not create spurious patterns. The average values of the non-pedestrian coefficients are near 1 since these are random images that do not share any common pattern. The pedestrian averages, on the other hand, show a clear pattern. The table shows strong response (values over 1.5) in the coefficients corresponding to the sides of the body. Conversely, the coefficients along the center of the body are very weak (values less than 0.5).

1.18	1.14	1.16	1.09	1.11
1.13	1.06	1.11	1.06	1.07
1.07	1.01	1.05	1.03	1.05
1.07	0.97	1.00	1.00	1.05
1.06	0.99	0.98	0.98	1.04
1.03	0.98	0.95	0.94	1.01
0.98	0.97	0.96	0.91	0.98
0.98	0.96	0.98	0.94	0.99
1.01	0.94	0.98	0.96	1.01
1.01	0.95	0.95	0.96	1.00
0.99	0.95	0.92	0.93	0.98
1.00	0.94	0.91	0.92	0.96
1.00	0.92	0.93	0.92	0.96

(a)

0.62	0.74	0.60	0.75	0.66
0.76	0.92	0.54	0.88	0.81
1.07	1.11	0.52	1.04	1.15
1.38	1.17	0.48	1.08	1.47
1.65	1.27	0.48	1.15	1.71
1.62	1.24	0.48	1.11	1.63
1.44	1.27	0.46	1.20	1.44
1.27	1.38	0.46	1.34	1.27
1.18	1.51	0.46	1.48	1.18
1.09	1.54	0.45	1.52	1.08
0.94	1.38	0.42	1.39	0.93
0.74	1.08	0.36	1.11	0.72
0.52	0.74	0.29	0.77	0.50

(b)

Table 1: Normalized vertical coefficients of scale 32×32 of images with (a) random natural scenes (without people), (b) pedestrians.

To visualize the emerging patterns for the different classes of coefficients we can use gray level to code the values of the coefficients and display them in the proper spatial layout. Each coefficient is displayed as a small square where coefficients close to 1 are gray, stronger coefficients are darker, and weaker coefficients are lighter. Figures 4(a)-(d) show the gray level coding for the arrays of coarse scale coefficients (32×32) and Figures 4(e)-(g) show the arrays of coefficients of the finer scale, (16×16).

Figure 4(a) shows the vertical coefficients of random images; as expected this figure is uniformly gray. The corresponding images for the horizontal and corner coefficients, not shown here, are similar. In contrast, the coefficients of the people, Figures 4(b)-(d), show clear patterns. It is interesting to observe that each class of wavelet coefficients is tuned to a different type of structural information. The vertical wavelets, Figure 4(b), capture the sides of the pedestrians. The horizontal wavelets, Figure 4(c), respond to the line from shoulder to shoulder and to a weaker belt line. The corner wavelets, Figure 4(d), are tuned better to corners, for example, the shoulders, hands and feet. The wavelets of finer scale in Figures 4(e)-(g) provide better spatial resolution of the body's overall shape and smaller scale details such as the head and extremities appear clearer. We conduct a similar analysis with the wavelets of the log of the intensities (that are related

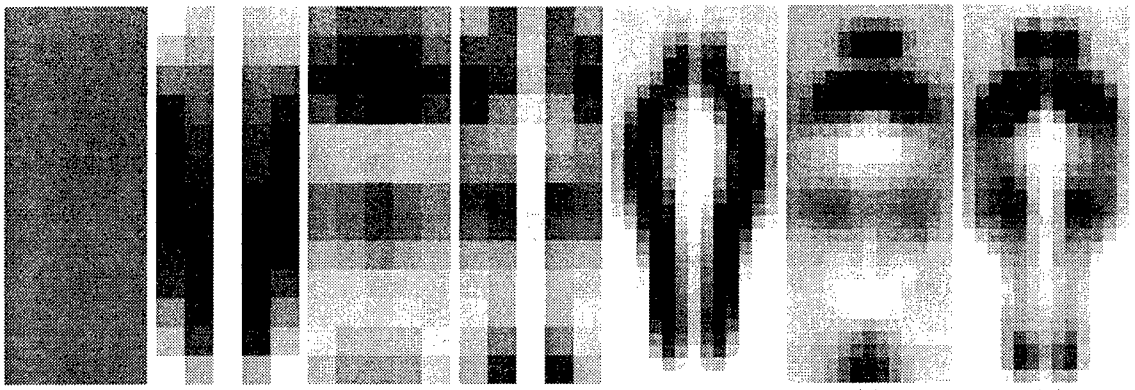


Figure 4: Ensemble average values of the wavelet coefficients coded using gray level. Coefficients whose values are above the template average are darker, those below the average are lighter. (a) vertical coefficients of random scenes. (b)-(d) vertical, horizontal and corner coefficients of scale 32×32 of images of people. (e)-(g) vertical, horizontal and corner coefficients of scale 16×16 of images of people.

to the ratio of intensities). The results of this statistical analysis are similar to the intensity differencing wavelets, indicating that, for pedestrians, the difference and ratio versions capture essentially identical information. An analysis using the sigmoid function as a "soft threshold" on the normalized coefficients yields equivalent results. In general, the learning of the coefficients can be based on different statistical analyses of the ensemble coefficients. We find it intriguing that a basic measure like the ensemble average provides clear identification of the coefficients as shown in Figure 4. The result of the analysis described above is a set of 29 coefficients that are consistent along the ensemble either as indicators of "change" or "no-change". There are 6 vertical and 1 horizontal coefficients at the scale of 32×32 and 14 vertical and 8 horizontal at the scale of 16×16. The identified set of coefficients is used as a feature vector for a classification algorithm which is trained to classify pedestrians from non-pedestrians.

We have decomposed the learning of the pedestrian class into a two-stage learning process. In the first stage, described in this section, we perform a dimensionality reduction where we identify the most important coefficients from the original set of 1326 wavelets coefficients (three types in two scales). Based on our initial experiments, it is doubtful that successful learning of the relationship between coefficients' values could be achieved on the original set of 1326 coefficients without introducing several orders of magnitude of additional training data. Most of these coefficients do not necessarily convey relevant information about the pedestrian class but, by starting with a large overcomplete dictionary, we would not sacrifice details or spatial accuracy. The above learning step extracts the most prominent features and results in a significant dimensionality reduction.

3 The Detection System

Once we have identified the important basis functions we can use various classification techniques to learn the relationships between the wavelet coefficients that define the pedestrian class. In this section, we present the overall architecture of the detection system, the classifier we used (the support-vector machine), and the training process. We conclude with experimental results of the detection system.

3.1 System Architecture

The system detects people in arbitrary positions in the image and in different scales. To accomplish this task, the system is trained to detect a pedestrian centered in a 128×64 pixel window. This training stage is the most difficult part of the system training and once it is accomplished the system can detect pedestrians at arbitrary positions, by scanning all possible locations in the image by shifting the 128×64 window. This is combined with iteratively resizing the image to achieve multi-scale detection. For our experiments, we scale the novel image from 0.2 to 1.5 times its original size, at increments of 0.1. At any given scale, instead of recomputing the wavelet coefficients for every window in the image, we compute the transform for the whole image and do the shifting in the coefficient space. A shift of one coefficient in the finer scale corresponds to a shift of 4 pixels in the window and a shift in the coarse scale corresponds to a shift of 8 pixels. Since most of the coefficients in the wavelet template are at the finer scale (the coarse scale coefficients hardly change with a shift of 4 pixels), we achieve an effective spatial resolution of 4 pixels by working in the wavelet coefficient space.

3.2 System Training

To train our system, we use a database of frontal and rear images of people from outdoor and indoor scenes. The initial non-people in the training database are patterns from natural scenes not containing people. The combined set of positive and negative examples form the initial training database for the classifier. A key issue with the training of detection systems is that, while the examples of the target class, in this case pedestrians, are well defined, there are no typical examples of non-pedestrians. The main idea in overcoming this problem of defining this extremely large negative class is the use of "bootstrapping" training [Sung and Poggio-1994]. After the initial training, we run

the system over arbitrary images that do not contain any people. Any detections are clearly identified as false positives and are added to the database of negative examples and the classifier is then retrained with this larger set of data. These iterations of the bootstrapping procedure allows the classifier to construct an incremental refinement of the non-pedestrian class until satisfactory performance is achieved. This bootstrapping technique is illustrated in Figure 5.

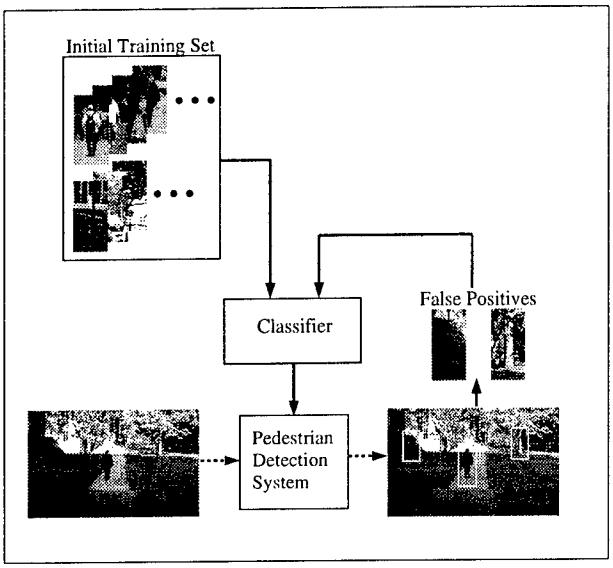


Figure 5: Incremental bootstrapping to improve the system performance.

3.3 Classification Schemes

As described in the previous section, the decision task, whether a given window contain a pedestrian or not, is the most difficult task and crux of the detection system. In Section 2.2.1 we describe the identification of the significant coefficients which characterized the pedestrian class. These coefficients can be used as feature vector for various classification methods.

3.3.1 Basic Template Matching

The simplest classification scheme is to use a basic template matching measure. As in Section 2.2.1, the normalized template coefficients are divided into two categories: coefficients above 1 (indicating strong change) and below 1 (weak change). For every novel window, the wavelet coefficients are compared to the pedestrian template. The matching value is the ratio of the coefficients in agreement. A similar approach was used in [Sinha-1994a] for face detection with good results. While this basic template matching scheme is very simple — better classification techniques can be applied — it is interesting to see how well it will perform on this more complex task.

3.3.2 Support Vector Machines

Instead of the simple template matching paradigm we can use a more sophisticated classifier which will learn the relationship between the coefficients from given sets of positive and negative examples. The classifier can learn more refined relationships than the simple template matching scheme and therefore can provide more accurate detection.

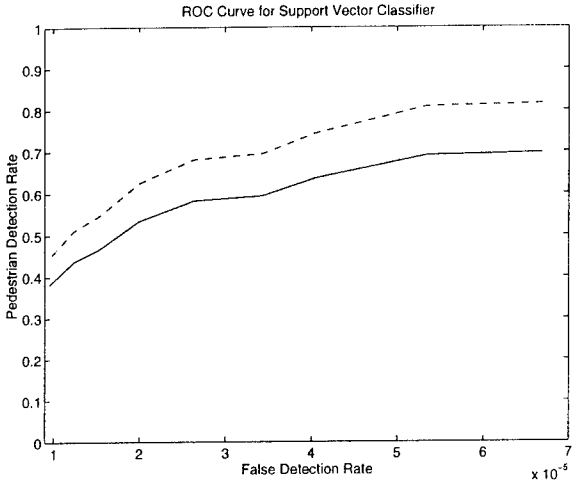


Figure 7: ROC curves for the support vector detection system; the bottom curve is over the entire test set, the top curve is over the "high quality" set.

The classification technique we use is the support vector machine (SVM) developed by Vapnik et al.[Boser *et al.*-1992][Vapnik-1995]. This recently developed technique has several features that make it particularly attractive. Traditional training techniques for classifiers, such as multilayer perceptrons (MLP), use empirical risk minimization and only guarantee minimum error over the training set. In contrast, the SVM machinery uses structural risk minimization which minimizes a bound on the generalization error and therefore should perform better on novel data. Another interesting aspect of the SVM is that its decision surface depends only on the inner product of the feature vectors. This leads to an important extension since we can replace the Euclidean inner product by any symmetric positive-definite kernel $K(x, y)$ [Riesz and Sz.-Nagy-1955]. This use of a kernel is equivalent to mapping the feature vectors to a high-dimensional space, thereby significantly increasing the discriminative power of the classifier. For our classification problem, we find that using a polynomial of degree two as the kernel provides good results.

It should be observed, that from the view point of the classification task, we could use the whole set of coefficents as a feature vector. However, using all the wavelet functions that describe a window of 128×64 pixels would yield vectors of very high dimensionality, as we mentioned earlier. The training of a classifier with such a high dimensionality would in turn require too large an example set. The template learning stage of Section 2.2.1 serves to select the basis functions relevant for this task and to reduce their number considerably (to a very reasonable 29).

4 The Experimental Results

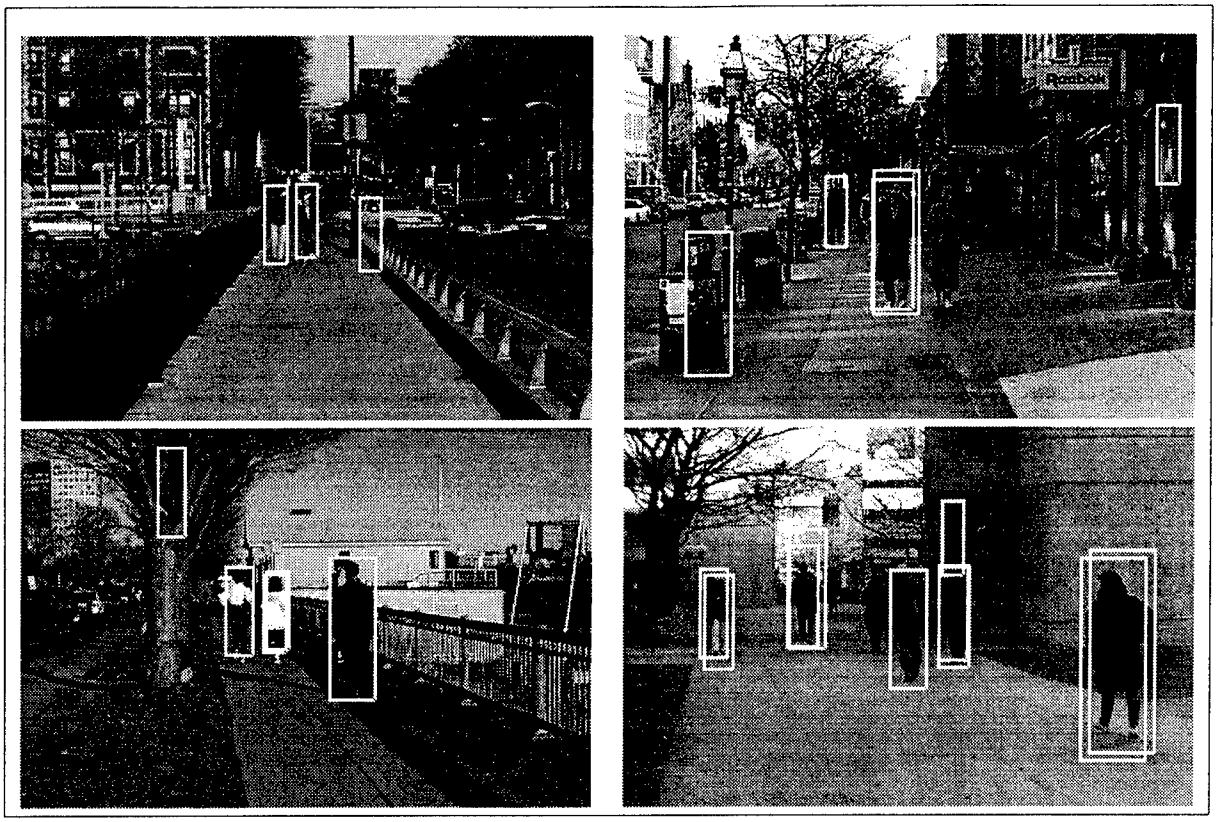


Figure 6: Results from the pedestrian detection system. These are typical images of relatively complex scenes that are used to test the system.

To evaluate the system performance, we start with a database of 564 positive examples and 597 negative examples. The system then undergoes the bootstrapping cycle detailed in Section 3.2. For this paper, the support vector system goes through three bootstrapping steps, ending up with a total of 4597 negative examples. For the template matching version a threshold of 0.7 (70% matching) was empirically found to yield good results.

Out-of-sample performance is evaluated over a test set consisting of 72 images for both the template matching scheme and the support vector classifier. The test images contain a total of 165 pedestrians in frontal or near-frontal poses; 24 of these pedestrians are only partially observable (e.g. with body regions that are indistinguishable from the background). Since the system was not trained with partially observable pedestrians, we would not even expect a perfectly trained system (with the current template) to detect these instances. To give a fair account of the system, we present statistics for both the total set and the set of 141 "high quality" pedestrian images. Results of the tests are presented in Table 2 for representative systems using template matching and support vectors.

The template matching system has a pedestrian detection rate of 52.7%, with a false positive rate of 1 for every 5,000 windows examined. The success of such a straightforward template matching measure suggests that the template learning scheme extracts non-trivial structural regularity within the pedestrian class.

	Detection Rate	*False Positive Rate (per window)*
Template Matching	52.7% (61.7%)	1:5,000
SVM	69.7% (81.6%)	1:15,000

Table 2: Performance of the pedestrian detection system; values in parentheses are for the set of "high quality" pedestrian images.

For the more sophisticated system with the support vector classifier, we perform a more thorough analysis. In general, the performance of any detection system exhibits a tradeoff between the rate of detection and the rate of false positives. Performance drops as we impose more stringent restrictions on the rate of false positives. To capture this tradeoff, we vary the sensitivity of the system by thresholding the output and evaluate the ROC curve, given in Figure 7. From the curve, we can see, for example, that if we have a tolerance of one false positive for every 15,000 windows examined, we can achieve a detection rate of 69.6%, and as high as 81.6% on the "high quality" set. As we expect, the support vector classifier with the bootstrapping training performs better than the "naive" template matching scheme.

In Figure 6 we show typical images that are used to test the system. These are very cluttered scenes crowded with complex patterns. Considering the complexity of these scenes and the difficulties of

pedestrian detection in natural outdoor scenes, we consider the above detection rate to be high. It is interesting to observe that most of the false positives are patterns with overall proportions similar to the human body. We believe that additional training and refinement of the current system will reduce the false detection rate further.
The system is currently trained only on frontal and rear views of pedestrians. Training the classifier to handle side views can be accomplished in an identical manner and is our next extension to the system.

5 Conclusion

In this paper, we introduce the idea of a redundant wavelet representation and demonstrate how it can be learned and used for pedestrian detection in a cluttered scene. This representation yields not only a computationally efficient algorithm but an effective learning scheme as well. The success of the wavelet representation for pedestrian detection comes from its ability to capture high-level knowledge about the object class (structural information expressed as a set of constraints on the wavelet coefficients) and incorporate it into the low-level process of interpreting image intensities. Attempts to directly apply low-level techniques such as edge detection and region segmentation are likely to fail in the type of images we analyze since these methods are not robust, are sensitive to spurious details, and give ambiguous results. Using the wavelet template, only significant information that characterizes the object class — as obtained in the learning phase — is evaluated and used.
The strength of our system comes from the expressive power of the redundant wavelet representation – this representation effectively encodes the intensity relationships of certain pattern regions that define a complex object class. The encouraging results of our system and related work on face detection, [Sinha-1994a] [Sinha-1994b], suggest that the approach described in this paper may well generalize to several other object detection tasks.

References

[Boser *et al.*, 1992] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optim margin classifier. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 144–52. ACM, 1992.

[Chen and Shirai, 1994] H-J. Chen and Y. Shirai. Detecting multiple image motions by exploiting temporal coherence of apparent motion. *Computer Vision and Pattern Recognition*, pages 899–902, 1994.

[Edgar Osuna and Girosi, 1996] Robert Freund Edgar Osuna and Federico Girosi. Support vector machines: Training and applications. *MIT CBCL-Memo*, May 1996. In preparation.

[Jacobs *et al.*, 1995] C.E. Jacobs, A. Finkelstein, and D.H. Salesin. Fast multiresolution image querying. *SIGGRAPH95*, August 1995. University of Washington, TR-95-01-06.

[Leung and Yang, 1987a] M.K. Leung and Y-H. Yang. Human body motion segmentation in a complex scene. *Pattern Recognition*, 20(1):55–64, 1987.

[Leung and Yang, 1987b] M.K. Leung and Y-H. Yang. A region based approach for human body analysis. *Pattern Recognition*, 20(3):321–39, 1987.

[Mallat, 1989] S.G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–93, July 1989.

[Moghaddam and Pentland, 1995] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. Technical Report 326, Media Laboratory, Massachusetts Institute of Technology, 1995.

[Riesz and Sz.-Nagy, 1955] F. Riesz and B. Sz.-Nagy. *Functional Analysis.* Ungar, New York, 1955.

[Rohr, 1993] K. Rohr. Incremental recognition of pedestrians from image sequences. *Computer Vision and Pattern Recognition*, pages 8–13, 1993.

[Rowley *et al.*, 1995] H.A. Rowley, S. Baluja, and T. Kanade. Human face detection in visual scenes. Technical Report CMU-CS-95-158, School of Computer Science, Carnegie Mellon University, July/November 1995.

[Sinha, 1994a] P. Sinha. Object Recognition via Image Invariants: A Case Study. In *Investigative Ophthalmology and Visual Science*, volume 35, pages 1735–1740. Sarasota, Florida, May 1994.

[Sinha, 1994b] Pawan Sinha. Qualitative image-based representations for object recognition. *MIT AI Lab-Memo*, No. 1505, 1994.

[Stollnitz *et al.*, 1994] E.J. Stollnitz, T.D. DeRose, and D.H. Salesin. Wavelets for computer graphics: A primer. *University of Washington, TR-94-09-11*, September 1994.

[Sung and Poggio, 1994] K-K. Sung and T. Poggio. Example-based learning for view-based human face detection. A.I. Memo 1521, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, December 1994.

[Tsukiyama and Shirai, 1985] T. Tsukiyama and Y. Shirai. Detection of the movements of persons from a sparse sequence of tv images. *Pattern Recognition*, 18(3/4):207–13, 1985.

[Vaillant *et al.*, 1994] R. Vaillant, C. Monrocq, and Y. Le Cun. Original approach for the localisation of objects in images. *IEE Proc.-Vis. Image Signal Processing*, 141(4), August 1994.

[Vapnik, 1995] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, 1995.

Self-Taught Visually-Guided Pointing for a Humanoid Robot

M. Marjanović B. Scassellati M. Williamson
Massachusetts Institute of Technology Cambridge MA

Abstract

The authors implemented a system which performs a fundamental visuomotor coordination task on the humanoid robot Cog. Cog's task was to saccade its pair of two degree-of-freedom eyes to foveate on a target, and then to maneuver its six degree-of-freedom compliant arm to point at that target. This task requires systems for learning to saccade to visual targets, generating smooth arm trajectories, locating the arm in the visual field, and learning the map between gaze direction and correct pointing configuration of the arm. All learning was self-supervised solely by visual feedback. The task was accomplished by many parallel processes running on a seven processor, extensible architecture, MIMD computer.

1 Introduction

This paper is one of a series of developmental snapshots from the Cog Project at the MIT Artificial Intelligence Laboratory. Cog is a humanoid robot designed to explore a wide variety of problems in artificial intelligence and cognitive science [5]. To date our hardware systems include a ten degree-of-freedom upper-torso robot, a multi-processor MIMD computer, a video capture/display system, a six degree-of-freedom series-elastic actuated arm, and a host of programming language and support tools [3, 4]. This paper focuses on a behavioral system that learns to coordinate visual information with motor commands in order to learn to point the arm toward a visual target. Additional information on the project background can be found in [5, 7, 14, 10].

To achieve visually-guided pointing, we construct a system that first learns the mapping from camera image coordinates $\vec{x} = (x, y)$ to the head-centered coordinates of the eye motors $\vec{e} = (pan, tilt)$ and then to the coordinates of the arm motors $\vec{\alpha} = (\alpha_0 ... \alpha_5)$. An image correlation algorithm constructs a saccade map $\vec{S} : \vec{x} \rightarrow \vec{e}$, which relates positions in the camera image with the motor commands necessary to foveate the eye at that location. Our task then becomes to learn the ballistic movement mapping from head-centered coordinates $\vec{e}$ to arm-centered coordinates $\vec{\alpha}$. To simplify the dimensionality problems involved in controlling a six degree-of-freedom arm, arm positions are specified as a linear combination of basis posture primitives. The ballistic mapping $\vec{B} : \vec{e} \rightarrow \vec{\alpha}$ is constructed by an on-line learning algorithm that compares motor command signals with visual motion feedback clues to localize the arm in visual space.

[0]The authors receive support from a National Science Foundation Graduate Fellowship, a National Defense Science and Engineering Graduate Fellowship, and JPL Contract # 959333, respectively. Support for the Cog project is provided by an ONR/ARPA Vision MURI Grant (No. N00014-95-1-0600), a National Science Foundation Young Investigator Award (No. IRI-9357761) to Professor Lynn Andrea Stein, and by the J.H. and E.V. Wade Fund.

2 Robot Platform

This section gives a brief specification of the physical subsystems of Cog (see Figure 1) that are directly relevant to our pointing task.

To approximate human eye movements, the camera system has four degrees-of-freedom consisting of two active "eyes" [1]. Each eye can rotate about a vertical axis (pan) and a horizontal axis (tilt). Camera images are digitized to produce 120×120 images in 8-bit grayscale.

The arm is loosely based on the dimensions of a human arm, and is illustrated in Figure 1. It has 6 degrees-of-freedom, each powered by a DC electric motor through a series spring (a series elastic actuator, see [13]). Motion of the arm is achieved by changing the equilibrium positions of the joints, not by commanding the joint angles directly.

The computational control for Cog is split into two levels: an on-board local motor controller for each motor, and a scalable MIMD computer that serves as Cog's "brain." This division of labor allows for an extensible and modular computer while still providing for rapid, local motor control. Each motor has its own dedicated local motor controller, a special purpose board with a Motorola 6811HC11E2 microcontroller, which reads the encoder, performs servo

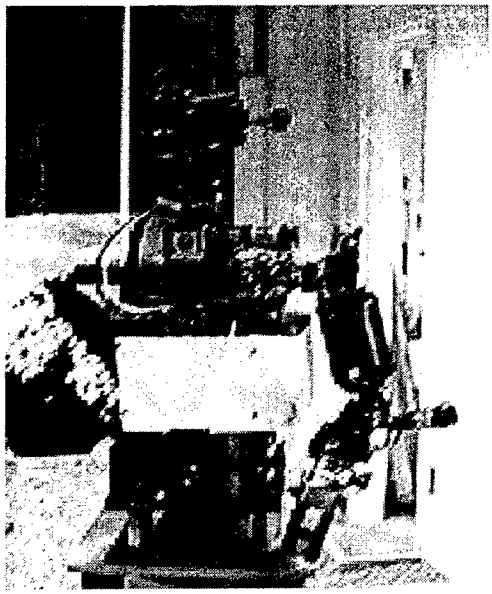


Figure 1: Cog, an upper-torso humanoid robot. Cog has two degrees-of-freedom in the waist, one in the shoulder, three in the neck, six on the arm, and two for each eye.

calculations, and drives the motor with a 32KHz pulse-width modulated signal. Cog's "brain" is a scalable MIMD computer consisting of up to 239 processor nodes (although only eight are in use so far). During operation, the brain is a fixed topology network. However, the topology can be changed and scaled by adding additional nodes and connections. All components of the processing system communicate through 8K by 16 bit DPRAM connections, so altering the topology is relatively simple.

3 Task Overview

The implementation discussed here can be decomposed into three major pieces, each developed semi-independently: visual, arm motor, and a ballistic map. The visual system is responsible for moving the eyes, detecting motion, and finding the end of the arm. The arm motor system maintains the variable-compliance arm and generates smooth trajectories between endpoints specified in a space of basis arm postures. The ballistic mapping system learns a feed-forward map from gaze position to arm position and generates reaching commands. Each of these subsystems is described in greater detail below.

Although the basic activity for this particular task is sequential — foveate, reach, train, repeat — there is no centralized scheduler process. Rather, the action is driven by a set of triggers passed from one process to another. This is not a very important design consideration with the single task in mind; however as we add more processes, which act in parallel and compete for motor and sensor resources, a distributed system of activation and arbitration will become a necessity.

4 Visual System

The components of the visual system used in this task can be grouped into three functional units: a saccade map trainer, a motion detection module, and a motion segmentation module.

4.1 Learning the Saccade Map

The saccade trainer incrementally learns the mapping between the location of salient stimuli in the visual image with the eye motor commands necessary to foveate on that object. With the neck in a fixed position, this task simplifies to learning the mapping between image coordinates and the pan/tilt encoder coordinates of the eye motors. The behavioral correlate of this simplified task is to learn the pan and tilt positions necessary to saccade to a visual target. Initial experimentation revealed that for the wide-angle cameras, this saccade map is linear near the image center but rapidly diverged near the edges. An on-line learning algorithm was implemented to incrementally update an initial estimate of the saccade map by comparing image correlations in a local field. This learning process, the saccade map trainer, optimized a look-up table that contained the pan and tilt encoder offsets needed to saccade to a given image coordinate.

Saccade map training began with a linear estimate based on the range of the encoder limits (determined during calibration). For each learning trial, the saccade map trainer generated a random visual target location (x_t, y_t) and recorded the normalized image intensities $\bar{I}_t$ in a 16×16 patch around that point. The process then issued a saccade motor command using the current map entries. After the saccade, a new image $\bar{I}_n$ is acquired. The normalized 16×16 center of the new image is then correlated against the target image. Thus, for offsets x_0 and y_0, we sought to maximize the dot-product of the image vectors:

$$\max_{x_0,y_0} \left(\sum_i \sum_j \bar{I}_t(i,j) \cdot \bar{I}_n(x_0+i, y_0+j) \right) \quad (1)$$

Since each image was normalized, maximizing the dot product of the image vectors is identical to minimizing the angle between the two vectors. This normalization also gives the algorithm a better resistance to changes in background luminance as the

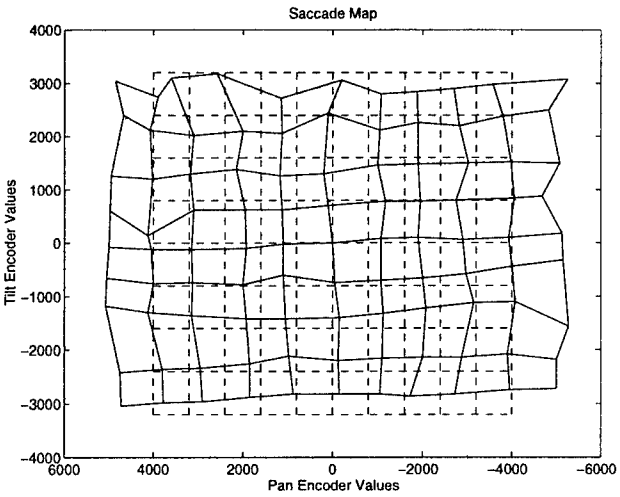


Figure 2: Saccade Map after 0 (dashed lines) and 2000 (solid lines) learning trials. The figure shows the pan and tilt encoder values for every tenth position in the image array within the ranges x=[10,110] (pan) and y=[20,100] (tilt).

camera moves. In our experiments, the offsets x_0 and y_0 had a range of $[-2, 2]$. The offset pair that maximized the expression in Equation 1, scaled by a constant factor, was used as the error vector for training the saccade map.

Note that a single learning step of this hill-climbing algorithm does not find the optimal correlations across the entire image. The limited search radius vastly increases the speed of each learning trial at the expense of producing difficulties with local maxima. However, in the laboratory space that makes up Cog's visual world, there are many large objects that are constant over relatively large pixel areas. The hill-climbing algorithm effectively exploited this property of the environment to avoid local maxima.

To simplify the learning process, we initially trained the map with random visual positions (x_t, y_t) that were multiples of ten in the ranges $[10, 110]$ for x_t (the pan dimension) and $[20, 100]$ for y_t (tilt). By examining only a subset of the image points, we could quickly train a limited set of points which would bootstrap additional points. Examining image points closer to the periphery was also unnecessary since the field of view of the camera was greater than the range of the motors; thus there were points on the edges of the image that could be seen but could not be foveated regardless of the current eye position. Figure 2 shows the data points in their initial linear approximation (dashed lines) and the resulting map after 2000 learning trials (solid lines). The saccade map after 2000 trials clearly indicates a slight counter-clockwise rotation of the mounting of the camera, which was verified by examination of the hardware. The training quickly reached a level of 1 pixel-error or less per trial within 2000 trials (approximately 20 trials per image location). Perhaps as a result of lens distortion effects, this error level remained constant regardless of continued learning.

Visual comparison of the target images before saccade and the new images after saccade showed good match for all training image locations after 2000 trials. A set of examples from the collected data is shown in Figure 3.

4.2 Motion Detection and Segmentation

The motion detection system uses local area differences between successive camera images to identify areas where motion has occured. The absolute value of the difference between the grayscale values in each image is thresholded to provide a raw motion image ($I_{raw} = \mathcal{T}(|I_0 - I_1|)$). The raw motion image is then used to produce a motion receptive field map, a 40×40 array in which each cell corresponds to the number of cells in a 3×3 receptive field of the raw motion image that are above threshold. This reduction in size allows for greater noise tolerance and increased processing speed.

The motion segmentation module takes the receptive field map from the motion detection processor and produces a bounding box for the largest contiguous motion group. The process scans the receptive field map marking all locations which pass threshold with an identifying tag. Locations inherit tags from adjacent locations through a region grow-and-merge procedure. Once all locations above threshold have been tagged, the tag that has been assigned to the most locations is declared the "winner". The bounding box of the winning tag is computed and sent to the ballistic map trainer.

5 Arm Motion Control

The method used to control the arm takes inspiration from work on organization of movement in the spinal cord of frogs [2, 8, 12]. These researchers electrically stimulated the spinal cord, and measured the forces at the foot, mapping out a force field in leg-motion space. They found that the force fields were convergent (the leg would move to fixed posture under the field's influence), and that there were only a small number of fields (4 in total). This lead to the suggestion that these postures were primitives that could be combined in different ways to generate movement [11]. Details on the application of this research to robotic arms can be found in [14].

In Cog's arm, the primitives are implemented as a

Figure 3: Expanded example of the visual learning of the saccade map. The center collage is the pre-saccade target images $\bar{I}_t$ for a subset of the entire saccade map. The left collage shows the post-saccade image centers with no learning. The right collage shows the post-saccade image centers after 2000 learning trials. The post-learning collage shows a much better match to the target than the pre-learning collage.

set of equilibrium angles for each of the arm joints, as shown in Figure 4. Each primitive corresponds to a different posture of the arm. Four primitives

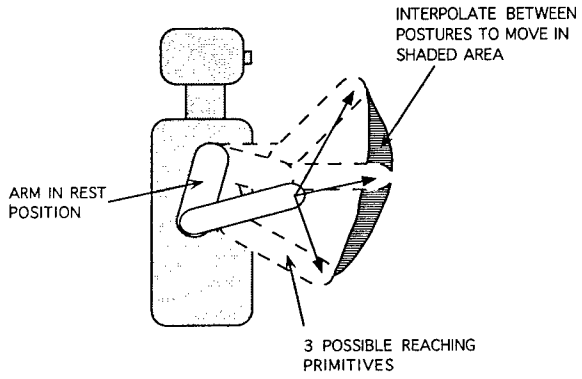


Figure 4: Primitives for the reaching task. There are four primitives: a rest position, and three in front of the robot. Linear interpolation is used to reach to points in the shaded area.

are used: a rest position, and three on the extremes of the workspace in front of the robot. These are illustrated in Figure 5. Positions in space can be reached by interpolating between the primitives, giving a new set of equilibrium angles for the arm, and so a new end-point position. The interpolation is linear in primitive and joint space, but due to the non-linearity of the forward kinematics (end-point position in terms of joint angles), the motion in Cartesian space is not linear. However since only 4 primitives are used to move the 6 DOF arm, there is a large reduction in the dimensionality of the problem, with a consequent reduction in complexity.

The reaching behavior takes inspiration from studies of child development [6]. Children always begin a reach from a rest position in front of their bodies. If they miss the target, they return to the rest position and try again. This reaching sequence is implemented in Cog's arm. Infants also have strong grasping and withdrawal reflexes, which help them interact with their environment at a young age. These reflexes have also been implemented on Cog.

6 Ballistic Map

The ballistic map is a learned function $\vec{B}$ mapping eye position $\vec{e}$ into arm position $\vec{\alpha}$, such that the resulting arm configuration puts the end of the arm in the center of the visual field. Arm position is specified as a vector in a space of three basic 6-dimensional joint position vectors — the *reach* primitives (shown in Figure 5). There is also a fourth "rest" posture to which the arm returns between reaches.

The reach primitive coefficents are interpreted as percentages, and thus are required to sum to unity. This constrains the reach vectors to lie on a plane, and the arm endpoint to lie on a two-dimensional manifold. Thus, the ballistic map $\vec{B}$ is essentially a function $\mathcal{R}^2 \rightarrow \mathcal{R}^2$.

We attempted to select reach primitives such that the locus of arm endpoints was smooth and 1-to-1 when mapped onto the visual field. The kinematics of the arm and eye specify a function $\vec{E} : \vec{\alpha} \mapsto \vec{e}$ which maps primitive-specified arm positions into the eye positions which stare directly at the end of the arm. The ballistic map $\vec{B}$ is essentially the inverse of $\vec{E}$: we desire $\vec{E}(\vec{B}(\vec{e})) = \vec{e}$. If $\vec{E}$ is 1-to-1, then $\vec{B}$ is single-valued and we need not worry about learning discontinuous or multiple output ranges.

The learning techniques used here closely parallels the distal supervised learning approach [9]. We actually learned the forward map $\vec{E}$ as well as $\vec{B}$; this was necessitated by our training scheme. However, $\vec{E}$ is useful in that it gives an expectation of where to look to find the arm. This can be used to generate a window of attention to filter out distractions in the motion detection.

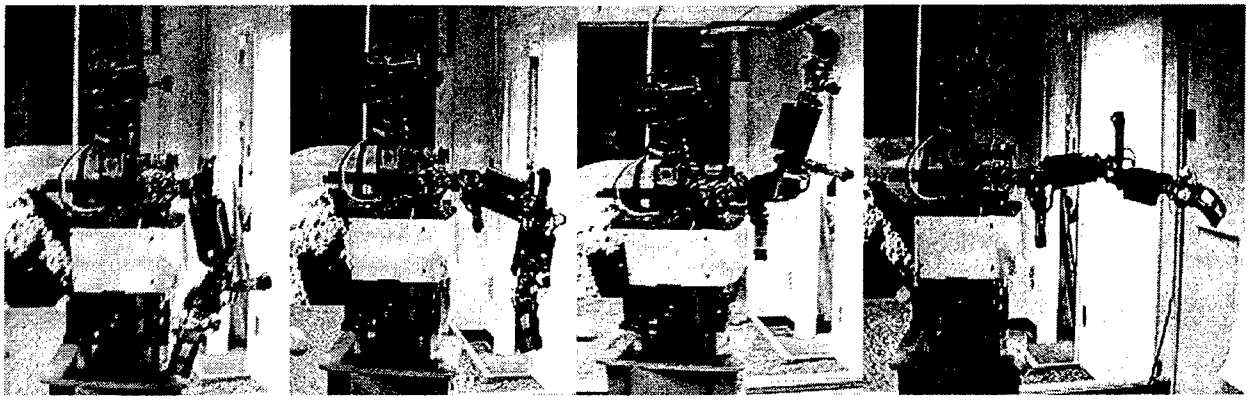


Figure 5: The basic arm postures. From left, "rest", "front", "up", and "side."

6.1 Map Implementation

The maps $\vec{B}$ and $\vec{E}$ are both implemented using a simple radial basis function approach. Each map consists of 64 Gaussian nodes distributed evenly over the input space. The nodes have identical variance, but are associated with different output vectors. The output of such a network ($\vec{y}$) for some input vector $\vec{i}$ is given by:

$$\vec{y} = \sum_k \vec{w}_k g_k(\vec{i}),$$

where

$$g_k(\vec{i}) = \exp(-\frac{1}{\sigma^2}\|\vec{i} - \vec{u}_k\|^2).$$

and $\vec{w}_k$ is a set of weights.

The ballistic map is initialized to point the arm to the center of the workspace for all gaze directions. The forward map is initialized to yield a centered gaze for all arm positions.

6.2 Learning the Ballistic Map

After the arm has reached out and its endpoint has been detected in the visual field, the ballistic map $\vec{B}$ is updated. However, since the error signal is a position in the image plane, the training cannot be done directly. We need to use the forward map $\vec{E}$ and the saccade map $\vec{S}$.

The current gaze direction $\vec{e}_0$ is fed through $\vec{B}$ to yield a reach vector $\vec{\beta}$ (β-space is a two dimensional parameterization of the α reach-primitive space). This $\vec{\beta}$ is sent to the arm to generate a reaching motion. It is also fed through the forward map $\vec{E}$ to generate an estimate $\vec{e}_p$ of where the arm will be in gaze-space after the reach. In an ideal world, $\vec{e}_p$ would equal $\vec{e}_0$.

After the arm has reached out, the motion detection determines the position $\vec{x}$ of the arm in pixel coordinates. If the reach were perfect, this would be the center of the image. Using the saccade map $\vec{S}$, we can map the difference in image (pixel) offsets between the end of the arm and the image center into gaze (eye position) offsets. So, we can use $\vec{S}$ to convert the visual position of the arm $\vec{x}$ into a gaze direction error $\Delta\vec{e}$.

We still cannot train $\vec{B}$ directly, since we have an e-space error but a β-space output. However, we can backpropagate $\Delta\vec{e}$ through the forward map $\vec{E}$ to yield a useful error term.

After all is said and done, we are performing basic least-mean-squares (LMS) gradient descent learning on the gaze error $\Delta\vec{e}$. For $\vec{B}$ defined by:

$$\vec{\beta} = \vec{B}(\vec{e}) = \sum_k \vec{w}_k g_k(\vec{e})$$

the update rule for the weights $\vec{w}_k$ is:

$$\Delta w_{ik} = -\eta \left(\Delta\vec{e} \cdot \frac{\partial \vec{F}}{\partial \beta_i} \right) g_k(\vec{e}).$$

for some learning rate η.

The forward map $\vec{F}$ is learned simultaneously with the ballistic map. Since $\vec{e} = \vec{e}_0 + \Delta\vec{e}$ is the gaze position of the arm after the reach, and $\vec{e}_p$ is the position predicted by $\vec{F}$, $\vec{F}$ can be trained directly via gradient descent using the error $(\vec{e}_p - \vec{e})$.

7 Results, Future Work, and Conclusions

At the immediate time of this writing, the complete system has been implemented and debugged, but has

not been operational long enough to fully train the ballistic map. Initial results on small subsets of the visual input space show promising results. However, it will take some more extended training sessions before Cog has fully explored the space of reaches.

In addition to completing Cog's basic ballistic pointing training, our plans for upcoming endeavors include:

- incorporating additional degrees of freedom, such as neck and shoulder motion, into the model
- refining the arm finding process to track the arm during reaching
- expanding the number of primitive arm postures to cover a full three-dimensional workspace
- extracting depth information from camera vergence and stereopsis, and using that to implement reaching to and touching of objects.
- adding reflexive motions such as arm withdrawal and a looming response, including raising the arm to protect eyes and head
- making better use of the inverse ballistic map in reducing the amount of computation necessary to visually locate the arm.

This pointing task, albeit simple when viewed alongside the myriad complex motor skills of humans, is a milestone for Cog. This is the first task implemented on Cog which integrates major sensory and motor systems using a cohesive distributed network of processes on multiple processors. To the authors, this is a long-awaited proof of concept for the hardware and software which have been under development for the past two and a half years. Hopefully, this task will be a continuing part of the effort towards an artificial machine capable of human-like interaction with the world.

8 Acknowledgments

The authors wish to thank the members of the Cog group (past and present) for their continual support: Mike Binnard, Rod Brooks, Cynthia Ferrell, Robert Irie, Yoky Matsuoka, Nick Shectman, and Lynn Stein.

References

[1] D. Ballard. Behavioral constraints on animate vision. *Image and Vision Computing*, 7:1:3–9, 1989.

[2] Emilio Bizzi, Ferdinando A. Mussa-Ivaldi, and Simon F. Giszter. Computations underlying the execution of movement: A biological perspective. *Science*, 253:287–291, 1991.

[3] R. Brooks. L. Technical report, IS Robotics Internal Document, January 1996.

[4] R. Brooks, J. Bryson, M. Marjanovic, L. A. Stein, , and M. Wessler. Humanoid software. Technical report, MIT Artificial Intelligence Lab Internal Document, January 1996.

[5] R. Brooks and L. A. Stein. Building brains for bodies. *Autonomous Robots*, 1:1:7–25, 1994.

[6] Adele Diamond. *Development and Neural Bases of Higher Cognitive Functions*, volume 608, chapter Developmental Time Course in Human Infants and Infant Monkeys, and the Neural Bases, of Inhibitory Control in Reaching, pages 637–676. New York Academy of Sciences, 1990.

[7] Cynthia Ferrell. Orientation behavior using registered topographic maps. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB-96)*. Society of Adaptive Behavior, 1996.

[8] Simon F. Giszter, Ferdinando A. Mussa-Ivaldi, and Emilio Bizzi. Convergent force fields organized in the frog's spinal cord. *Journal of Neuroscience*, 13(2):467–491, 1993.

[9] M. I. Jordan and D. E. Rumelhart. Forward models: supervised learning with a distal teacher. *Cognitive Science*, 16:307–354, 1992.

[10] Matthew Marjanović, Brian Scassellati, and Matthew Williamson. Self-taught visually-guided pointing for a humanoid robot. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB-96)*. Society of Adaptive Behavior, 1996.

[11] Ferdinando A. Mussa-Ivaldi and Simon F. Giszter. Vector field approximation: a computational paradigm for motor control and learning. *Biological Cybernetics*, 67:491–500, 1992.

[12] Ferdinando A. Mussa-Ivaldi, Simon F. Giszter, and Emilio Bizzi. Linear combinations of primitives in vertebrate motor control. *Proceedings of the National Academy of Sciences*, 91:7534–7538, August 1994.

[13] Gill A. Pratt and Matthew M. Williamson. Series elastic actuators. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-95)*, volume 1, pages 399–406, Pittsburg, PA, July 1995.

[14] Matthew M. Williamson. Postural primitives: interactive behavior for a humanoid robot arm. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB-96)*. Society of Adaptive Behavior, 1996.

Performance and Human Interface Issues of a System for Visual Interpretation of Hand Gestures *

Rick Kjeldsen
IBM T.J.Watson Research Center
P.O.Box 704, Yorktown Hgts, NY

John R. Kender
Department of Computer Science
Columbia University, NY, NY

ABSTRACT

We review the design of a working system that visually recognizes hand gestures for the control of a window based user interface. We summarize many of the most significant environmental and human factors aspects of this modality of interaction that we determined, and indicate how they impact system design and utility. We explore two such interface issues in depth. First, we describe how it is necessary and possible to visually smooth the camera input using a non-linear physical model of the cursor. Second, we show how a standard HCI model of object selection (Fitts' Law) can be extended to model system visual tracking and visual selecting performance. We conclude by presenting and evaluating total system performance.

1 Introduction

We have previously described a working system that visually recognizes hand gestures for control of a window based user interface. Knowledge of natural human gesticulation is used to design the system for ease of use and ease of recognition. Task knowledge is made explicitly available to the system in the form of a grammar describing the interaction language. This provides a sense of context within the conversation, allowing the system to focus on relevant image events, and to use simple but potentially ambiguous features. We avoid the complexity of an internal model of hand shape by using a neural network to classify various poses. The result is a system which is easily modified to perform a range of similar tasks in various environments.

A camera placed below the screen captures a sequence of images of a gesturing hand against a relatively stationary background. The hand is segmented from low resolution decimations of the image sequence using color and various image processing operations. The size and location of the hand in each image are used to create a sequence of X-location, Y-location, Size (XYS) tokens, which is translated to screen coordinates, smoothed and used to position the cursor where the user is pointing on the screen. The motion of the hand is interpreted by extracting symbolic features, such as pauses, changes of direction, or distance from the sequence of smoothed tokens. The sequence of features is interpreted by traversing a transition network encoding the interaction language. At certain points in the network it becomes necessary to classify the pose of the hand. Then a high resolution image is cropped tightly around the region indicated by the current XYS token and a more accurate segmentation is performed. This image is preprocessed to a canonical form and passed to a neural net that has been trained to differentiate the various hand poses. The classification is used to determine the next node in the transition network. At key points in the interpretation of a gesture, the transition network calls out actions that do such things as bring up menus, or select or move objects on the screen. (For more details of system operations, see [Kjeldsen, 1996].)

2 Human Interface Issues

In this section, we present some high-level observations on how the gesture understanding system's physical setup and environment interact with what appear to be natural human characteristics and preferences. We also indicate how a simple visual add-on to an existing mouse-based interface cannot be fully successful, as the two modalities differ substantially in their understanding of the interactions.

*This work is supported in part by DARPA contract DACA-76-92-C-007.

2.1 Environmental Considerations

2.1.1 Physical Setup

Placing the camera below the screen and looking up avoids extreme foreshortening of the arm and makes it easy to detect a gesture when it raises off the keyboard to gesture. Although bare forearms are a problem, the hand looms large in the image making it easy to find, and other skin blobs, including the user's face, are guaranteed to be much smaller. Even though the typical placement of camera for video-conferencing is above the screen, for both these applications the ideal camera placement is actually inside the screen.

The user does not appear to think in terms of the location of their hand in the image, but rather with respect to the screen. The system creates the illusion that the fingertip is a laser pointer and the cursor appears where the beam would contact the screen; this illusion is non-linear, as by geometry it must vary with the vertical location on the screen.

2.1.2 Lighting

Human skin has a distinct color, rich in lightly saturated red tones, a characteristic that persists across a wide range of apparent skin colors and lighting conditions. The fundamental assumption of the segmentation method is that skin coloration is relatively unique in the target environment. But we also note that people rarely paint walls ceilings with skin-tone paints or wear skin-tone clothes, as important objects like people need to stand out in the environment, for human purposes as well.

Experimentation has determined that the skin is a highly Lambertian surface, and that office lighting is usually diffuse for many humanly desirable reasons. Consequently, attempting to only use the intensity of the color signal results in a washed out image of the hand. However, the normalized red channel of the color signal is actually brighter in shadows, and subtracting it from the intensity channel helps highlight features of the hand, including creases and the lines between fingers.

Generally, the human need for strong office lighting ensures enough image contrast, and the properties of skin reduce color variations due to specularity, light source coloration, inter-reflections, and surface normal direction. We have found that color segmentation generalizes across different skin-tones, including people of Asian, Indian, European, and African descent, with the major exception being that training on Asian tones does not seem to generalize as well to others, and vice versa.

2.1.3 Physics, Gravity, and Geometry

The cursor and other screen objects have been loosely modeled as masses to be dragged by springs. However, the spring function is non-linear, in order to tradeoff between making the cursor movement feel too sluggish at moderately large movements, and too twitchy at small displacements. (We detail this concept in the next section.) Similarly, altering object mass dynamically adjusts the "feel" of the objects appropriately. The mass of the cursor is increased when moving a window, so that the user gets the feeling that the object is heavier, and it lags noticeably behind the hand; however, it is now quite stable when the hand is not moving. Thus, weighty windows can be positioned accurately, whereas by the same methods, moderately heavy menus can be selected accurately and quickly, and lightweight cursors can be tracked very rapidly.

Vertical positioning appears more difficult than horizontal positioning, as vertical positioning requires extending the hand away from the body with large muscle groups and fighting gravity, while horizontal positioning only requires rotating the shoulder or wrist. However, these preferences are in contradiction to standard "pull-down" menus.

Users tend not to deliberately accommodate the imaging geometry. Instead, once users are not constrained by the keyboard, they tend to lean way back in the chair or lean very close to the screen, but rarely sit up as they would when typing. Also, some people tend to gesture with their hand far from the screen, causing it to image relatively small, and making it difficult to set a size threshold. Further, the pointing pose in particular varies greatly depending on where on the screen the user is pointing; this requires additional care in the neural net.

2.2 Human Aspects

2.2.1 Gestural Communication

Although not formal, human gestural thought and communication induce certain constraints on the visual signal that the system exploits. People naturally use gestures that are easy to differentiate based only on appearance; further, they display them in such a

way that the important features are visible and sometimes exaggerated. Hand models are therefore less important, and appearance matching suffices. Some poses are very close to each other in joint-angle space but have different "meaning": for example, middle and index fingers extended, but the two fingers are together or apart. The reverse is also true, for example, grasping, but either a grape or a grapefruit.

The system exploits the natural "reset" signal: at any time, the user can drop the hand to the keyboard and the system will return to the start state. This is a very useful and naturally elicited gesture when the system misinterprets a user's intended gesture.

People tend to adopt gestural shorthand. The "comma" is a unique shorthand gesture that we discovered. It is defined as the user moving their hand smoothly back away from the screen then toward it again. It is used to end one phrase of the gestural sentence and begin another: for example, to select the window they are pointing to, and then to point at it again to move it. It is also easily detected, as both a decrease in size and an apparent downward image movement. When we allowed the user to move and resize a window repeatedly, separating commands with commas, this proved to be very usable and reliable.

The interaction style used here was based on current theories of natural gesticulation [Quek, 1993]. They observe that a gesture has three phases: preparation, stroke, and retract (the PSR cycle). However, as the language evolved and incorporated modifications for usability and practicality, it moved away from such a pure grammar; a good example is the comma, which allows users to string together commands without returning to the keyboard between them. After some experience, then, we gave interaction language a simple sentence-like structure of subject-verb-adverb; most adverbs tend to be analog, so using a gesture is natural to express them. Further, we allowed compounding, ellipsis, and anaphora, with the comma being literally equivalent to a comma. If the user returns to typing on the keyboard, the elided verb is "select", for example, "That window [select]". If the user gestures a comma and then another verb-like pose, they are anaphorically repeating the subject, for example, "That window select, [that window] resize like this, [that window] move here."

2.2.2 Gestural Rhythm

Gestures appear to have their own inherent rhythms. Tracking requires a system response of no more than 10 hz, which is the sampling rate for the most rapid human pointing movements of 5 hz; we also observed that below 5 hz a user feels that movement is artificially slow. Our system at 7-8 hz therefore was adequate, until the user to became expert and moved quickly. Conversely, some poses must be held till the system responds; our system recognized poses at about 2 hz, which was annoying but acceptable. We noted that slow tracking and pose analysis interacted adversely: too long a pause disturbed the rhythm of the gesture, and made subsequent motions more awkward and harder to interpret.

We observed that the rhythm of fine positioning seems to vary somewhat from person to person and from time to time: the user either uses slow deliberate motions, or a move-wait-move pattern. Both are equally difficult to detect precisely.

2.2.3 Training of New Users

New users are different from more expert ones. A user often shapes the hand awkwardly for training, and then points more naturally while selecting a window, and pose recognition performance then drops to around 80%. Eventually, however, the system tends to train the user: the user learns to provide inputs in such a way that the system recognizes it correctly. This is analogous to what happens with mouse actions, such as the proper timing of a double click. Nevertheless, with no memory aids for what actions are permitted next, some users complained that it was difficult to learn. And, since a pause is a significant gesture, uncertainty in gesture often lead to an unintended gestural sentence.

2.3 A New Interface Model

Standard HCI techniques must be modified to handle hand gesture interaction. The flexibility and continuous nature of gesture, and the expense involved in extracting gestural features, make hand gestures fundamentally different from most current interface devices. Hands never turn off; gesture positioning is inherently three dimensional; gestures can deliver multiple pieces of information at one time; and it is possible to combine multiple poses by having one pose change into another.

Many of the problems which were encountered were

due to aspects of the original window system GUI which were retained, which has evolved to suit the characteristics of a keyboard and mouse. Mouse positioning is precise and easy, but the set of gestures (button clicks) is information poor; as a result, current GUIs rely on accurate selection of numerous small icons. Similarly, pull-down menus are comfortable with a mouse, but unnatural using free-hand gesture; horizontal or pie-wedge menus would be more appropriate. Nevertheless, performance of the two modalities can be both measured and modeled using similar tools, and found to be approximately equivalent. (We detail this concept in a separate section.)

3 Paradoxical Cursor Tracking

We determined that the smoothing constraints of image and cursor tracking are critically dependent on the context in which hand motion occurs. Some of these constraints appear paradoxical. The displayed cursor must remain relatively still when the hand is not intentionally moving, but the cursor must respond to small movements almost immediately so that fine positioning is possible. It is also important that the cursor not lag far behind when the hand is moving quickly so that the user can select an object rapidly, but the cursor must not overshoot or oscillate when the motion of the hand suddenly stops.

3.1 Non-Linear Springs

Modeling the cursor as a physical object with mass, position, and velocity meets some but not all of these criteria. Instead, the system detects the various contexts, and the physical model is adjusted dynamically depending on apparent user intention. The force function, which transmits visual location to cursor position via a sigmoidally varying nonlinear spring, depends on current and prior positions and velocities. The figures show the force curves for various values of initial velocity (Figure 1). The net effect is that the cursor always moves toward the hand, but never passes it. When the hand has been moving slowly, additional small movements cause the cursor to gently accelerate toward it, while large movements cause the cursor to fly after it and catch up quickly. Behavior changes smoothly between these extremes. When the hand has been moving rapidly, the cursor tracks any displacement closely.

Experience and analysis indicate that three possible scenarios should govern this dynamic adjustment:

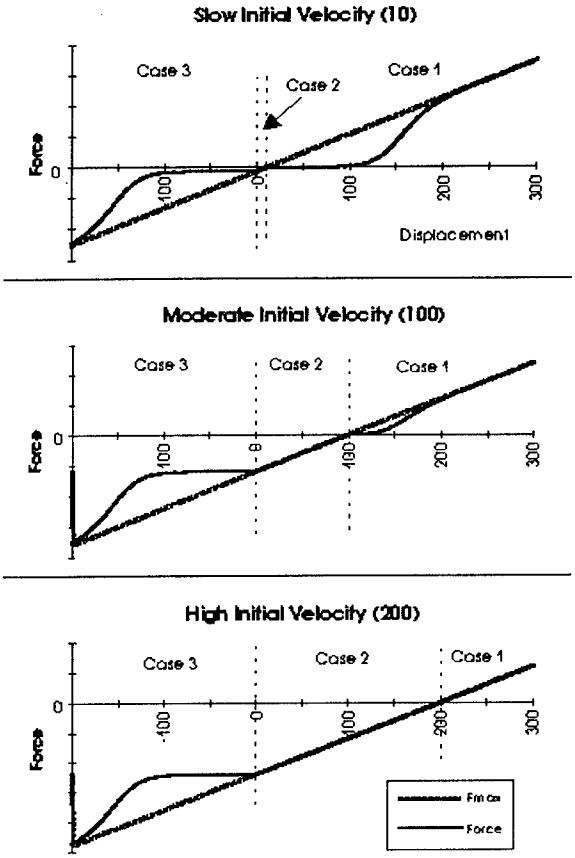


Figure 1: Force applied to the cursor versus hand displacement for three initial velocities. A force of Fmax gets the cursor exactly to the current hand position by the end of the current cycle.

Case 1: At the current velocity the cursor cannot not catch up with the new hand position in one cycle.

Case 2: At the current velocity, the cursor will catch up to or pass the new hand position in this cycle.

Case 3: The current velocity of the cursor is carrying it away from the hand.

Figure 1 shows force curve for various values of initial velocity. In case 3, the behavior is independent of initial velocity; the cursor always is given enough force to stop, then heads toward the hand with an acceleration that depends on cursor to token displacement. In case 2 the cursor always decelerates, so that it meets the hand this cycle. The behavior in case 1 depends strongly on initial velocity. When displacement is small, the cursor chases the hand as per the sigmoid function, but at high initial velocity it follows the hand as in case 2, with intermediate velocities producing behavior between these extremes.

3.2 Types of Image Noise

The noise in the raw XYS token stream has very distinctive characteristics. The noise can be subdivided into a stable component and a time-varying component. The stable component consists of an constant offset from the ideal cursor location, caused by local segmentation errors, but when the hand is still it remains constant. The varying components of the noise only show up when a sequence of images are considered. Some is again caused by random errors in the segmentation and small unintentional movements of the hand. Analysis of the sources of the remainder indicate that it falls roughly into four categories: 1) low-frequency step or ramp functions due to the hand passing in front of widely different backgrounds; 2) medium-frequency medium-amplitude variations due to chaotic segmentations of fingers on or off the hand; 3) high-frequency low-amplitude due to jitter from segmentation effects of region border pixels; 4) sporadic high-amplitude jumps due to image blur during fast motion.

Except for the second category, which depends critically on fine tuning of the sigmoid function, these noise types are well handled by the non-linear smoothing.

4 Modeling and Measuring Performance

The system has been used by the first author during development and testing for hundreds of hours. It has also been used by about a dozen others for shorter amounts of time during demos and formal testing. Since it is as portable as any PC, it has been used in several different office and lab environments, without need for controlled lighting, prepared backgrounds, or unusual care in its placement. It is not being used on a daily basis primarily because of its limited functionality.

Quality of the segmentation varies depending on quality of calibration, environmental conditions, and the orientation/position of the hand within the scene. At its best, 15-20% of the time, segmentation of the hand is near perfect. But more likely, 80-85% of the time, the segmentation is less than perfect but accurate enough for smooth tracking and reliable pose recognition. Occasionally, less than 1% of the time, large chunks of the hand are missing, it may break into many small pieces, or some other object than the hand is segmented instead.

4.1 Left-Right Selection

As an initial test, we studied the absolute accuracy and repeatability of visual tracking in an alternating target task, that is, left-right hand moton. The smoothing algorithm effectively damps out the majority of the jitter that is present in the raw hand position data, but nevertheless tracks fast movements very well.

4.2 Hand vs. Mouse

Next, we compared the usability of the visual interface with that of standard pointing and clicking in several ways. First, we evaluated the ability of users to select on-screen objects by measuring the time it takes to place the cursor in an object on the screen. For objects larger than about an inch, selection time was found to be comparable to that when using a mouse.

Selection time was measured from the moment the space key was pressed with the pointing hand to until the cursor had been inside a one-inch target continuously for 0.5 seconds. The mean selection time for free-hand pointing was 1.91 seconds; the mean selection time using the mouse was 1.57. These times include the 0.5 seconds within the target. However, free-hand pointing time drops rapidly with increasing target size, leveling out at around 1.2 seconds when the target reaches 2 inches in diameter; selection time with a mouse also drops, to about 1.3 seconds.

4.3 Fitts' Law Extended

Secondly, we modeled free-hand pointing according to an augmented version Fitts' Law, a well-used and well-verified predictor of the time needed for object selection for a wide range of pointing devices and selection tasks. Fitts' Law basically states that the time T to select an object of a given size W at distance D from the initial location of the cursor is proportionate to log(D/W), suitably scaled and translated with three task-dependent constants:

$$T = a + b * \log(D/W + c).$$

Using first the literature, then our own data to tune the parameters, we found that the pure form of Fitts' Law accurately captured our mouse data. But following the same process with the hand data, it was not possible to come up with reasonable values of the constants to make the model explain the increase in selection time for small objects.

When terms were added for lag and noise, however, a reasonable fit was possible, as Figure 2 shows. Following the literature, the scale parameter, b, was increased to account for the lag in system response caused by slow tracking rate. Following our data, the shift parameter, a, was increased to account for jitter; assuming that jitter is random noise following a normal distribution, through straightforward probability theory we derived a closed form for the jitter-induced delay for any target size and expected amplitude of noise. Together, these two models then allowed us to predict both systems' performance as a function of tracking rate and tracking accuracy.

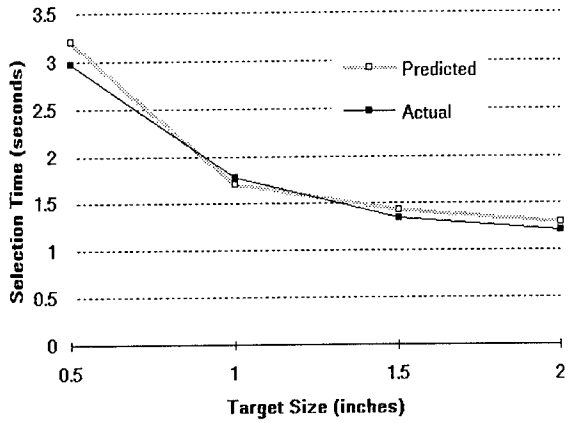


Figure 2: Predicted and actual selection time for targets of various sizes using free-hand pointing.

The model shows that random jitter in the cursor position and the lag caused by the slow tracking rate are sufficient to cause the long selection times for small objects. The model indicated that accuracy was far more critical than speed, particularly since the target had to be selected and held for a fixed amount of time. With very little noise and at a tracking speed attainable with off-the-shelf hardware in a few years, free-hand pointing can be expected to be approximately the same as for a mouse, slightly better for large objects, and slightly worse for small ones. Under ideal conditions (i.e. no tracking lag at all), gesture has the potential to be significantly faster than using a mouse for objects of all sizes.

4.4 Total System Performance

Lastly, we measured the total system results for two experienced users. In the following table, the qualified success rate indicates how many times the action was performed essentially correctly, but there was a minor error, such as the window ending up in the wrong position. The pure success rate indicates how often the action was performed exactly as the user intended. The minor errors nearly always have a common cause, namely trouble detecting where the retraction movement began, which in turn is primarily due to the relatively slow tracking speed.

Task	Qualified Success	Pure Success
Select	94%	89%
Move Window	85%	81%
Resize Window	87%	84%
Select Menu Item	82%	63%

Selection, the simplest task, was also the most reliable. Moving a window or resizing it were also reliable, and the similar performance results cross-validate each other. Bringing up a menu is reliable, but selecting the very small items from it is not, at least not yet. For further discussion, please see [Kjeldsen and Kender, 1997].

5 Conclusion

We have designed a working system capable of controlling window manipulation in a user interface using hand gestures. This has made it possible to study the interaction of the environment with natural human preferences, to experiment with different hand gesture interaction styles and algorithms, and to measure model the component and systems performance. It has also indicated that for full success, a simple visual gesture system add-on to a mouse-based design is inadequate; the full interface has to accommodate the unique needs of human gestural communication.

References

[Kjeldsen and Kender, 1997] R. Kjeldsen and J.R. Kender. Context-based visual hand gesture recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, To appear, June 1997.

[Kjeldsen, 1996] R. Kjeldsen. *Visual Interpretation of Hand Gestures as a Practical Interface Modality*. PhD thesis, Department of Computer Science, Columbia University, October 1996.

[Quek, 1993] F. Quek. Hand gesture interface for human-machine interaction. In *Proceedings of Virtual Reality Systems*, Fall 1993.

PNF Propagation and the Detection of Actions Described by Temporal Intervals

Claudio Pinhanez Aaron Bobick
Perceptual Computing Group
MIT Media Laboratory
20 Ames St. - Cambridge, MA 02139
pinhanez,bobick@media.mit.edu

Abstract

This paper presents a method for the representation and detection of action using Allen's temporal algebra and a time propagation algorithm based on a novel method called the **PNF** propagation. In this scheme, an action is represented as a collection of time intervals corresponding to its sub-actions, events, and detectable states of physical objects. The paper provides the basics of the **PNF** propagation which extends Allen's interval algebra into handling causal propagation of the states of temporal intervals. Some examples and results are given at the end, showing that the technique works well when there are sufficiently strong causal links from detectors to actions.

1 Introduction

In this paper we present a method — the **PNF** propagation — for detection of actions represented as a collection of temporal intervals. **PNF** propagation expands Allen's interval algebra [Allen, 1984] into causal recognition of intervals and provides a representation of the state of temporal intervals upon which fast algorithms can be used.

The aim of this paper is to present the **PNF** propagation and how it can be used in the recognition of human actions. We start by discussing the problem of representing human action and how Allen's interval algebra can be used to model the underlying temporal structure. Following, we introduce the concept of **PNF**-restriction, the basic component of **PNF** propagation. **PNF**-restriction can be approximately computed by using a technique for simplifying constraint satisfaction problems — CSP — called arc-consistency ([Mackworth, 1977]). We conclude by analyzing some results obtained using this technique.

2 Actions and Intervals

Human actions typically decompose into smaller, simpler units whose occurrence can be verified empirically by sensors attached to a machine. We believe that the methods for action recognition proposed before [Nagel, 1995, Kaultz and Allen, 1986, Siskind, 1994] are unable to cope with most of the complex time patterns of everyday actions which include external events, simultaneous activities, and multiple sequencing possibilities [Allen and Ferguson, 1994].

```
Intervals
 { pick-up-bowl
   reach-for-bowl grasp-bowl
   bowl-out-of-hands bowl-in-hands
   DET:hands-close-sta-bowl
   DET:bowl-on-table DET:bowl-off-table }

Relations
 { { reach-for-bowl pick-up-bowl { START } }
   { grasp-bowl pick-up-bowl { FINISH } }
   { reach-for-bowl grasp-bowl { MEET BEFORE } }
   { reach-for-bowl bowl-out-of-hands { DURING FINISH } }
   { bowl-out-of-hands bowl-in-hands { MEET iMEET } }
   { grasp-bowl bowl-in-hands { MEET OVERLAP } }
   { DET:hands-close-sta-bowl bowl-out-of-hands
    { START EQUAL DURING FINISH } }
   { DET:bowl-on-table bowl-out-of-hands
    { START DURING FINISH EQUAL } }
   { DET:bowl-off-table bowl-in-hands
    { START DURING FINISH EQUAL } } }
```

Figure 1: Representation of a "pick-up bowl" action using temporal intervals.

To deal with such structures we propose to use a representation based on Allen's *interval temporal logic* [Allen, 1984] which explicitly incorporates multiple temporal possibilities for action structures. Allen's interval algebra is based on the 13 possible primitive relationships between two time intervals: the equality **EQUAL**, the relations **BEFORE**, **MEET**, **OVERLAP**, **DURING**, **START**, **FINISH**, and their inverses, **iBEFORE**, **iMEET**, **iOVERLAP**, **iDURING**, **iSTART**, and **iFINISH** (see [Allen, 1984] for the definition of those relationships).

Figure 1 shows the representation for the temporal structure of a "pick-up bowl" action as it is used by

the detection system described later in this paper. The interval `pick-up-bowl` corresponds to the period where the action of picking up the bowl is occuring. This action is decomposed into two sub-actions corresponding to the intervals `reach-for-bowl` and `grasp-bowl`.

The relations between those three intervals are defined by the first three lines of the "Relations" section. `reach-for-bowl` is declared to be a time interval which has the same beginning as `pick-up-bowl`, but finishes first — the `START` relationship. Similarly, `grasp-bowl` finishes at the same time as `pick-up-bowl`. The relationship between `reach-for-bowl` and `grasp-bowl` is defined in more vague terms: they either immediately follow each other (`MEET`) or happen in sequence (`BEFORE`). In other words, there can be some time between the sub-actions corresponding to pauses and indecisions. This example illustrates how modeling the temporal structure using Allen's logic allows the representation of the indeterminacy and the multiple possibilities of typical of human actions.

The next level of decomposition involves two mutually exclusive predicates (written as `MEET` or `iMEET`) about the physical relation between the bowl and the hands, `bowl-in-hands` and `bowl-out-of-hands`. The fact that reaching for the bowl must happen while the bowl is not in contact with the hands is expressed by the `DURING` or `FINISH` relationship between `reach-for-bowl` and `bowl-out-of-hands`. Similarly, `bow-in-hands` starts during `grasp-bowl` or just after its end.

For each of the two "physical" predicates we can assign simple, low-level detectors (marked by the prefix `DET:`). The first, `DET:hands-close-sta-bowl`, detects if the hands are close to the bowl while the bowl is static and on the table. The other two detectors `DET:bowl-on-table` and `DET:bowl-off-table` identify the presence of the bowl on or off the table. The first two detectors can fire only when the bowl is out of hands, while `DET:bowl-off-table` can only happen while the bowl is being held.

Notice that most of the relationships defined in this example do not involve "deep" commonsense reasoning. For instance, `bowl-in-hands` and `bowl-out-of-hands` are temporally mutually exclusive simply because by definition they represent different states for the bowl. But by preprocessing the representation using Allen's time constraint propagation algorithm ([Allen, 1984]), we can obtain — and use — stricter temporal constraints imposed by the structure of the action. For instance, in the case described above Allen's algorithm propagates the fact that `reach-for-bowl` is followed by `grasp-bowl` into detecting that `bowl-out-of-hands` must `MEET` `bowl-in-hands` — instead of the original `MEET` or `iMEET`.

As we can see, there are many advantages in representing action structure through temporal intervals and their relationships. The next section describes a theory and an algorithm which enable the detection of the occurrence of actions and sub-actions given the state of the "detector" intervals.

$f(\lambda, r)$: possible states of I_B, given $I_A\, r\, I_B$			
	state λ of I_A		
r	past	now	fut
EQUAL	past	now	fut
BEFORE	past/now/fut	fut	fut
iBEFORE	past	past	past/now/fut
MEET	past/now	fut	fut
iMEET	past	past	now/fut
OVERLAP	past/now	now/fut	fut
iOVERLAP	past	past/now	now/fut
START	past/now	now	fut
iSTART	past	past/now	fut
DURING	past/now	now	now/fut
iDURING	past	past/now/fut	fut
FINISH	past	now	now/fut
iFINISH	past	now/fut	fut

Table 1: A restriction function $f(\lambda, r)$.

3 PNF-Restriction

Action detection using `PNF` propagation employs an algorithm, called `PNF`-*restriction*, which we developed based on algorithms for CSP. This section explains and defines `PNF`-restriction, and provides algorithms to compute it.

3.1 The Basic Technique

`PNF`-restriction is based on assigning labels from the set $m = \{\texttt{past}, \texttt{now}, \texttt{fut}\}$ to intervals to intuitively capture the idea of an interval that happened in the "past", is happening "now", or can happen in the "future". If a label $l \in m$ is assigned to an interval, we say that the *state* of the that interval is l.

The basic idea of our action detection method is to assign `now` values to low-level detectors and run an algorithm that determines the possible states of the other intervals by considering the temporal constraints between the actions and the detectors. The process of generating the possible states of all intervals, given the states of some of them, is called `PNF`-*restriction*.

Considering a set of intervals and the temporal relationships between any two of them, there can be assignments of labels to intervals which violate the intuitive meaning of "past", "now", and "future" occurrences. For instance, if an interval I_A happens strictly `BEFORE` interval I_B, both intervals can not be happening at the same time.

Using our label assignment, this is translated to the condition that if I_A is `BEFORE` I_B, and I_B is known to be in the state `now`. To not violate temporal constraints, I_A must be in the `past` state. However, if we consider the opposite situation where it is known that the value of I_A is `past`, then the value of I_B is not constrained in any sense: I_B may have already happened (`past`), or be happening (`now`), or be in the future (`fut`).

In general, given the state of an interval I_A, λ, and a particular relationship $r \in \mathcal{A}$ to another interval I_B,

we can define a *restriction function* f which produces the possible states $f(\lambda, r)$ of the interval I_B. Table 1 displays a particular $f(\lambda, r)$ which embeds the usual semantics of past, present, and future among time intervals. The multiple options for I_B are necessary because there are situations where the state of I_A does not fully determine the state of I_B, as it was the case shown in the previous example.

3.2 Component PNF-states

To represent the multiple possibility of assignment of values from m to the intervals we develop the concept of *component* **PNF**-*states*. Consider the set M of subsets of m,

$$M = \{\emptyset, \{\texttt{past}\}, \{\texttt{now}\}, \{\texttt{fut}\}, \{\texttt{past}, \texttt{now}\}, \{\texttt{past}, \texttt{fut}\}, \{\texttt{now}, \texttt{fut}\}, \{\texttt{past}, \texttt{now}, \texttt{fut}\}\}$$

whose elements are abbreviated as

$$M = \{\emptyset, \texttt{P}, \texttt{N}, \texttt{F}, \texttt{PN}, \texttt{PF}, \texttt{NF}, \texttt{PNF}\}$$

Given an interval, we can assign one of the members of M to represent its admissible states. This is called the **PNF**-*state* of the interval. To represent the **PNF**-state of all intervals, we define the set U of n-tuples on M where each n-tuple W is referred as a *component* **PNF**-*state*,

$$W = (W_1, W_2, \ldots, W_n) \text{ where each } W_i \in M$$

and $W \in U = \prod^n M$. When one of the components of a component **PNF**-state W is the empty set, we say that W is a *collapsed* component **PNF**-state, or simply, a *collapsed state*. To simplify notation, we sometimes write $(W_1, W_2, \ldots, W_n)$ as $(W_i)_i$.

It is useful to define a containment relation between two component **PNF**-states of U, $W = (W_i)_i$ and $Y = (Y_i)_i$, by making $W \subseteq Y$ only if $W_i \subseteq Y_i$, for all $i = 1, 2, \ldots, n$. Similarly, we define an algebra[1] on U by the union and intersection functions, $W \cup Y = (W_i \cup Y_i)_i$ and $W \cap Y = (W_i \cap Y_i)_i$.

A component **PNF**-state $W = (W_i)_i$ is called *simple* if each W_i is an unitary set, $W_i \in \{\texttt{P}, \texttt{N}, \texttt{F}\}$. Intuitively, a simple component **PNF**-state represents a specific assignment of values from m to every interval.

Component **PNF**-states are useful for representing possible states of intervals, and not for individual assignments to the intervals. For instance, if a pair of intervals can only assume the values $(\texttt{P}, \texttt{N})$ and $(\texttt{N}, \texttt{P})$, the component representation which includes the two assignments, $(\texttt{PN}, \texttt{PN})$, also includes two other assignments which can not happen, $(\texttt{P}, \texttt{P})$ and $(\texttt{N}, \texttt{N})$. This problem is not important in our action detection method because we want to determine only the possible states of each interval, considered independently of the others.

Moreover, component **PNF**-states are a good notation to represent the input from the sensors. Typically, we associated to a binary sensor only two **PNF**-states, **N** if the sensor is true, and **PF** if the sensor is false. Therefore, we can represent all the information coming from sensors by a single component **PNF**-state $S = (S_i)_i$, where

$$S_i = \begin{cases} \text{value of the } I_i\text{–sensor} & \text{if } I_i \text{ is a sensor} \\ \texttt{PNF} & \text{otherwise} \end{cases}$$

3.3 Definition of PNF-Restriction

We are now in condition to formalize the basic algorithm underlying our action detection method. We start by expanding the definition of restriction function given in table 1 to component **PNF**-states. The **PNF**-*restriction function* $\mathcal{F} : M \times \mathcal{A} \rightarrow M$ is defined as the union of the corresponding f's, according to

$$\mathcal{F}(\Lambda, r) = \bigcup_{\lambda \in \Lambda} f(\lambda, r) \text{ where } \Lambda \in M \text{ and } r \in \mathcal{A}$$

Let us then consider a set R of interval relations between intervals in I, $R \subseteq I \times \mathcal{A} \times I$, and define R_{ij} as the set of all relations between intervals I_i and I_j which are members of R,

$$R_{ij} = \{r \in \mathcal{A} \mid (I_i, r_{ij}, I_j) \in R\}$$

Notice that R_{ij} is not necessarily a singleton because we may not know the exact relationship between the intervals I_i and I_j. The statement $R_{ij} = \{\texttt{BEFORE}, \texttt{MEET}\}$ informs that I_i is either before I_j *or* meets I_j.

Now, given a component **PNF**-state $W = (W_i)_i$ and a set of relations $R \subseteq I \times \mathcal{A} \times I$, W is said an R-*possible* component **PNF**-state of U if for every two intervals I_i and I_j,

$$W_j \subseteq \bigcup_{r \in R_{ij}} \mathcal{F}(W_i, r)$$

that is, possible component **PNF**-states are the ones which respect the time structure defined by f, and, as a consequence, our intuitive expectations about time.

We then define the *restriction of a component* **PNF**-*state* W *under* R, $\mathcal{R}(W)$, as the union of all simple, R-possible component **PNF**-states under R which are contained in W,

$$\mathcal{R}(W) = \bigcup_{\substack{Y \subseteq W, Y \text{ simple} \\ Y \ R\text{-possible}}} Y$$

Thus, if we apply the restriction function $\mathcal{R}$ on a component **PNF**-state S representing the information from the sensors as described above, we obtain the component **PNF**-state $\mathcal{R}(S)$ describing exactly the values that each interval can assume in at least one situation.

[1] To make notation simpler, we extend the usual meaning of the symbols $\subseteq$, $\cap$, and $\cup$ to the equivalent operations between component **PNF**-states.

3.4 PNF-Restriction as a CSP Problem

We can the computation of PNF-restriction as a constraint satisfaction problem ([Kumar, 1992, Nadel, 1989]). To do that, it is necessary to find a mapping between our formulation of PNF-restriction into a CSP, by determining the CSP's variables and the predicates corresponding to the unary and binary constraints. First, we interpret the n intervals $I_1, I_2, \ldots, I_n$ as variables with domain {P, N, F}. Doing this we implicitly state that all the solutions of the CSP are **simple** component PNF-states.

Based in the constraint function $\mathcal{F}$, given two intervals I_i and I_j and the relations between them, R_{ij}, we can define binary constraints between the variables based on predicates P_{ij} which are true only if the PNF-state W_j of I_j is compatible with the PNF-state W_i of I_i,

$$P_{ij} = \begin{cases} true & W_j \subseteq \bigcup_{r \in R_{ij}} \mathcal{F}(W_i, r) \\ false & \text{otherwise} \end{cases}$$

so any solution which satisfies those constraints can be interpreted as a R-possible component PNF-state.

The input component PNF-state W is interpreted as a set of unary constraints (in fact, disequations) on the variables of the associated CSP. For instance, if the input is a component PNF-state $(W_1, W_2, W_3) =$ (PNF, N, PF), it represents the predicates $W_2 \neq$ past, $W_2 \neq$ fut, and $W_3 \neq$ now. The unary constraints assure that the solutions are contained in the input W.

Therefore, the PNF-restriction of a component PNF-state W, $\mathcal{R}(W)$, can be computed by determining all solutions for the associated CSP, considering than as simple PNF-states, and taking their component-wise union. By construction all solutions of the CSP are simple, R-possible component PNF-states contained in W. Notice, however, that unlike in a traditional CSP the objective is not to enumerate all solutions individually, but to obtain the values for each interval which appear in at least one solution.

3.5 Algorithms for PNF-Restriction

As we see from the preceding paragraph, PNF-restriction can be computed by the search and component-wise union of all solutions in the associated CSP. However, finding one solution for a binary CSP is a NP-complete problem if the domain of each variable has more than two values (by reduction to the 3-SAT problem, [Downing and Gallier, 1984]). Therefore, it is possible that calculating the PNF-restriction is also a NP-complete problem.

A common technique to simplify CSPs is called *arc-consistency*, which requires polynomial time ([Mackworth, 1977]). We have been applying arc-consistency to calculate a component PNF-state which is guaranteed to contain the PNF-restriction. For practical purposes, the result of arc-consistency seems to produce all the information needed for action detection as we will show later with some experimental results.

Fig. 2 shows the arc-consistency algorithm ([Mackworth, 1977]) adapted to the component PNF-state notation. The first step of the algorithm consists in detecting which intervals of a component PNF-state W have a state different than PNF, and queue all those intervals for further expansion. Then $\overline{W}$, the variable used to construct the component PNF-state to be returned, is initialized identically to the input W.

```
Input:  I a set of intervals, I = {I_i};
        R the relations between the intervals,
          R ⊆ I × A × I
        F a component restriction function,
          F : M × A → M
        W a component PNF-state, W = (W_i);
Output: R(W), the PNF-propagation of W under R
Algorithm:
    initialize queue with all intervals I_i of I
      where W_i ≠ PNF                                  (1)
    W̄ ← W                                              (2)
    while queue ≠ ∅                                    (3)
        I_io ← pop(queue)                              (4)
        ( W̄_io denotes the state of I_io in W̄)         (5)
        for each interval I_i ∈ I,
          such as (I_io, r, I_i) ∈ R                   (6)
            X ← ⋃_{r ∈ R_io,i} F(W̄_io, r)              (7)
            when W̄_i ≠ W̄_i ∩ X                         (8)
                W̄_i ← W̄_i ∩ X                          (9)
                push(I_i, queue)                       (10)
    return W̄                                           (11)
```

Figure 2: An arc-consistency-based algorithm to compute an upper bound for the restriction $\mathcal{R}(W)$ of a component state W.

The core of the algorithm is a loop which ends when the queue is empty. In each cycle, one interval I_{i_o} at state $\overline{W}_{i_o}$ is examined. For each interval I_i to which I_{i_o} relates to, a variable X is assigned the union, for all the relations, of the results of the restriction function. In other words, the algorithm considers all the possible relationships between the two intervals, and considers the safest situation by taking the union. In the next step, if necessary, the component PNF-state is actualized with the intersection of the corresponding PNF-state and X, and the modified interval is pushed into the queue.

Since the result of arc-consistency always contains all the solutions of a CSP, it is clear that the PNF-restriction of a component PNF-state is contained in the component PNF-state computed by the arc-consistency algorithm of fig. 2. Moreover, in our use of that algorithm, we have never found a situation where the arc-consistency algorithm has produced a component PNF-state differently than PNF-restriction. In other words, we have some reason to believe that computing PNF-restriction is in fact a simpler problem than computing all the solutions of a CSP, and achievable in polynomial time by the arc-consistency algorithm of fig. 2. Presently we are trying to determine whether this conjecture is true.

4 Action Detection Using PNF Propagation

When using **PNF**-restriction for action detection, we consider the computed **PNF**-state of each interval to determine whether the action may be happening. If the **PNF**-state of the interval is **N**, we can say that the action is happening; if it is **P**, **F**, or **PF** the action can be said to be not happening; otherwise (**PNF**), we assign an indeterminate label.

PNF-restriction deals exclusively with determining a coherent structure for the world *at a given moment of time*. However, after an interval is determined to be in the **past**, its state for the rest of time is completely determined, i.e., **past**. To capture this concept, we define a function that time-expands a component **PNF**-state into another which is guaranteed to be true in the next moment of time.

4.1 Time Expansion

The objective is to define a *time expansion function*, $\mathcal{T} : U \rightarrow U$, that considers a component **PNF**-state W^t at time t and computes the smallest component **PNF**-state W^{t+1} at time $t+1$ that satisfies the intuitive meanings of past, now, and future.

We start by defining a time expansion function for each element of $m = \{\mathtt{past}, \mathtt{now}, \mathtt{fut}\}$, $\mathcal{T}_m : m \rightarrow M$. To preserve the intuitive meanings of past, present, and future, a natural choice is the time expansion function, $\mathcal{T}_m : m \rightarrow M$

$$\mathcal{T}_m(\mathtt{past}) = \mathtt{P} \quad \mathcal{T}_m(\mathtt{now}) = \mathtt{PN} \quad \mathcal{T}_m(\mathtt{fut}) = \mathtt{NF}$$

Given the function that time-expands elements $\mathcal{T}_m$, we define the function that expands the elements of M, $\mathcal{T}_M : M \rightarrow M$ as being the union of the results of $\mathcal{T}_m$,

$$\mathcal{T}_M(\Phi) = \bigcup_{\phi \in \Phi} \mathcal{T}_m(\phi)$$

and the time expansion $\mathcal{T} : U \rightarrow U$ as the component-wise application of $\mathcal{T}_M$ on a component **PNF**-state $W = (W_i)_i$, $\mathcal{T}(W) = (\mathcal{T}_M(W_1), \mathcal{T}_M(W_2), \ldots, \mathcal{T}_M(W_n))$.

4.2 PNF Propagation

PNF *propagation* is a method to detect the occurrence of actions which is based on intersecting the information from the sensors with the time expansion of the component **PNF**-state representing all the past information.

Formally, given the n time intervals corresponding to all actions, sub-actions, and detectors, **PNF** propagation determines the **PNF**-state of each interval at each time t, represented by the component **PNF**-state W^t. Sensor information at time t is gathered in the component **PNF**-state S^t where all states are **PNF** except those corresponding to perceptual sensors, which are assigned the sensor's corresponding **PNF**-state as explained before.

The initial component **PNF**-state W^0 represents an initial state of total ignorance, $W^0 = (\mathtt{PNF})_i$. After that, we determine W^t by computing the restriction on the intersection between the time expansion of W^{t-1} and S^t,

$$W^t = \mathcal{R}(\mathcal{T}(W^{t-1}) \cap S^t)$$

It is necessary to time expand the component W^{t-1} before intersecting it with the perceptual information S^t, since between instant $t-1$ and t actions may have ended or begun. Using past information is a fundamental component of the power of **PNF** propagation, as shown in the following experimental results.

4.3 Examples of Results

Figure 3 shows some results for the detection of the action "pick-up bowl". The top part of the figure displays the temporal diagrams for the **PNF**-state of the detectors (marked as **DET:**) and the true state of all other intervals (marked as **TRUE:**), for a particular instance of the action of picking up a bowl. The data were obtained manually from a video depicting the action. The diagram employs different symbols for each **PNF**-state, under the convention that the bottom line represents the **fut** state, the middle represents **now**, and the top, **past** (see the legend at the bottom of fig. 3).

Fig. 3a shows the results when the detection process uses the description of the "pick-up bowl" action exactly as given in fig. 1. Basically, only the physical events related to intervals **bowl-in-hands** and **bowl-out-of-hands** are recognized. The occurrence of the main action, **pick-up-bowl**, is never detected, although in the initial period it has been ruled out the possibility that the action had already happened (**NF**), and, after **DET: bowl-off-table** becomes **N**, it is detected that the action is happening or has already happened (**PN**).

The primary problem is that the original definition of "pick-up bowl" lacks a causal link between detecting the bowl is not on the table and the result of the act of grasping. Part *b* of fig. 3 shows that, with the addition of such relation, the end of **pick-up-bowl** is detected.

To detect the beginning of **pick-up-bowl**, it is necessary that the action description includes some causal relationship about the beginning of the sub-action **reach-for-bowl**. A possible way is to indicate that the proximity between hands and bowl (as detected by **DET:hands-close-sta-bowl**) is an indicator for the occurrence of **reach-for-bowl**. Notice that by doing this, we are assigning a relationship which may not be always true. However, given the simplicity of our sensors (and of most state-of-the-art vision-based algorithms), such "intentional" links are necessary to detect higher level actions. The results, shown in part *c*, display the almost perfect detection of **pick-up-bowl** and **reach-for-bowl**.

Finally, if we also want to detect the occurrence of **grasp-bowl**, a new detector is necessary. This is shown in part *d* of fig. 3, which displays the diagram of a new sensor, **DET:hands-touch-bowl** which fires precisely when the hand touches the bowl. In this

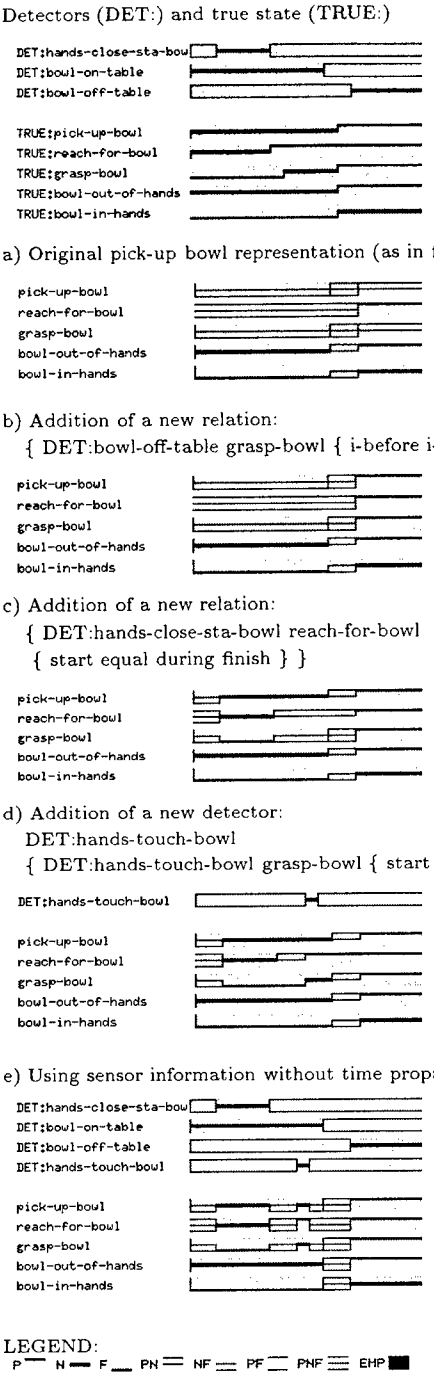


Figure 3: Detection of the action "pick-up bowl".

last case, the state of the intervals are known in most times, and are correct (compare to the **TRUE:** diagram at the top of the figure).

Part *e* of fig. 3 shows the importance of the information from the previous instant of time on the strength of **PNF** propagation. In this case, W^t is computed solely on the information from the sensors, $W^t = \mathcal{R}(S^t)$. Comparing fig. 3.*e* with part *d*, we can see a distinct degradation in the results. The main reason is that after a cause for an interval being in the **now** state ceases to exist, the system still

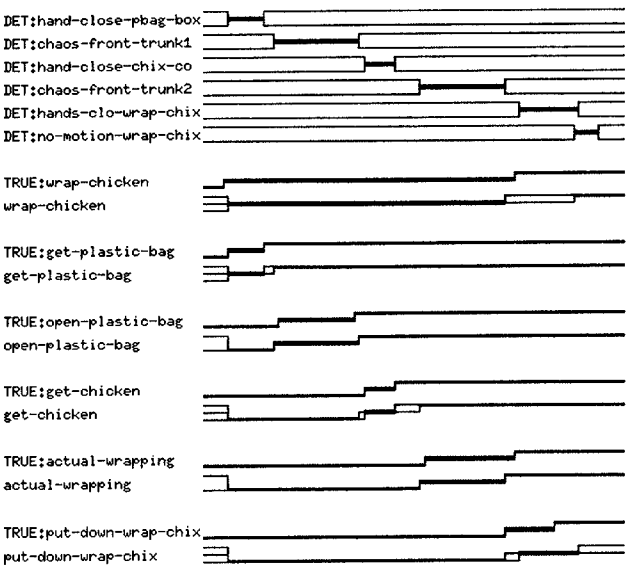


Figure 4: Detection of the action "wrapping chicken".

considers that the interval can still happen in the **fut** (compare **pick-up-bowl** in both cases).

Figure 4 illustrates the detection of a more complex action, wrapping chicken with a plastic bag which involves 25 intervals and 6 detectors. This action is derived from our previous work on automated cameras for TV cooking shows ([Pinhanez and Bobick, 1996]). The figure displays the true and the recognized state of the main action and of five sub-actions, which of them with a level of complexity similar to the "pick-up bowl" shown above. All the sensors are very simple: proximity between hands and the box containing plastic bags (**DET:hand-close-pbag-box**) and the plate containing chicken (**DET:hand-close-chix-co**), chaotic movement in front of the subject's trunk (**DET:chaos-front-trunk**), and absence of motion in the area of the wrapped chicken (**DET:no-motion-wrap-chix**).

5 Final Remarks

We have no knowledge of previous research trying to recognize actions defined by structures as loose as we are allowing in our experiments; most previous action recognition schemes [Siskind, 1994, Nagel, 1995] use strict sequential definitions of actions which do not reflect the way actions happen in everyday life.

In [Pinhanez and Bobick, 1996] we employed a variation of Schank's "conceptualization" ([Schank, 1975]) to decompose primary actions into sub-actions and physical events using a one pass inference algorithm. We believe that an extension of that method can produce loose temporal relationships between sub-actions such as those of fig. 1.

We are aware of some limitations of the approach. The first, obvious one, is if the computation of **PNF**-restriction is in fact exponential in time, that is, if we determined that the algorithm based on arc-

consistency is too weak for detection purposes. However, we do know from our tests that **PNF**-restriction actually reduces significantly a component **PNF**-state. A second limitation refers to the expressive capabilities of using intervals and their temporal relationships to represent human actions. For instance, in our work in *SingSong* [Pinhanez *et al.*, 1997], we realized the need of provisions to represent cyclic actions which are not met even by Allen's temporal logic. One possible approach to allow cycles is to modify the time expansion function of **past** to $\mathcal{T}_m(\mathbf{past}) = \mathbf{PNF}$.

References

[Allen and Ferguson, 1994] James F. Allen and George Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5):531–579, 1994.

[Allen, 1984] James F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.

[Downing and Gallier, 1984] William Downing and Jean J. Gallier. Linear time algorithms for testing the satisfiability of propositional horn formulae. *Journal of Logic Programming*, 1(3):267–284, 1984.

[Kaultz and Allen, 1986] Henry A. Kaultz and James F. Allen. Generalized plan recognition. In *Proc. of Fifth National Conference of Artificial Intelligence*, 1986.

[Kumar, 1992] Vipin Kumar. Algorithms for constraint-satisfaction problems: a survey. *AI Magazine*, 13:32–44, 1992.

[Mackworth, 1977] A. K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1):99–118, 1977.

[Nadel, 1989] Bernard A. Nadel. Constraint satisfaction algorithms. *Computational Intelligence*, 5:188–224, 1989.

[Nagel, 1995] Hans-Hellmut Nagel. A vision of 'vision and language' comprises action: An example from road traffic. *Artificial Intelligence Review*, 8:189–214, 1995.

[Pinhanez and Bobick, 1996] Claudio S. Pinhanez and Aaron F. Bobick. Approximate world models: Incorporating qualitative and linguistic information into vision systems. In *AAAI'96*, pages 1116–1123, Portland, Oregon, August 1996.

[Pinhanez *et al.*, 1997] Claudio S. Pinhanez, Kenji Mase, and Aaron F. Bobick. Interval scripts: A design paradigm for story-based interactive systems. In *Proc. of CHI'97*, Atlanta, Georgia, March 1997.

[Schank, 1975] Roger C. Schank. Conceptual dependency theory. In *Conceptual Information Processing*, chapter 3, pages 22–82. North-Holland, 1975.

[Siskind, 1994] Jeffrey Mark Siskind. Grounding language in perception. *Artificial Intelligence Review*, 8:371–391, 1994.

Omnidirectional Video Camera *

Shree K. Nayar

Department of Computer Science, Columbia University
New York, New York 10027
Email: nayar@cs.columbia.edu

Abstract

Conventional video cameras have limited fields of view that make them restrictive in a variety of vision applications. There are several ways to enhance the field of view of an imaging system. However, the entire imaging system must have a single effective viewpoint to enable the generation of pure perspective images from a sensed image. A new camera with a hemispherical field of view is presented. Two such cameras can be placed back-to-back without violating the single viewpoint constraint, to arrive at a truly omnidirectional sensor. Results are presented on the software generation of pure perspective images from an omnidirectional image, given any user-selected viewing direction and magnification. The paper concludes with a discussion on the spatial resolution of the proposed camera.

1 Introduction

Conventional imaging systems are quite limited in their field of view. Is it feasible to devise a video camera that can, at any instant in time, "see" in all directions? Such an *omnidirectional* camera would have an impact on a variety of applications, including autonomous navigation, remote surveillance, video conferencing, and scene recovery.

Our approach to omnidirectional image sensing is to incorporate reflecting surfaces (mirrors) into conventional imaging systems. This is what we refer to as *catadioptric* image formation. There are a few existing implementations that are based on this approach to image sensing (see [Nayar, 1988], [Yagi and Kawato, 1990], [Hong, 1991], [Goshtasby and Gruver, 1993], [Yamazawa *et al.*, 1995], [Nalwa, 1996]). As noted in [Yamazawa *et al.*, 1995] and [Nalwa, 1996], in order to compute pure perspective images from a wide-angle image, the catadioptric imaging system must have a single center of projection (viewpoint). In [Nayar and Baker, 1997], the complete class of catadioptric systems that satisfy the single viewpoint constraint is derived. Since we are interested in the development of a practical omnidirectional camera, two additional conditions are imposed. First, the camera should be easy to implement and calibrate. Second, the mapping from world coordinates to image coordinates must be simple enough to permit fast computation of perspective and panoramic images.

We begin by reviewing the state-of-the-art in wide-angle imaging and discuss the merits and drawbacks of existing approaches. Next, we present an omnidirectional video camera that satisfies the single viewpoint constraint, is easy to implement, and produces images that are efficient to manipulate. We have implemented several prototypes of the proposed camera, each one designed to meet the requirements of a specific application. Results on the mapping of omnidirectional images to perspective ones are presented. In [Peri and Nayar, 1997], a software system is described that generates a large number of perspective and panoramic video streams from an omnidirectional video input. We conclude with a discussion on the resolution of the proposed camera.

2 Omnidirectional Viewpoint

It is worth describing why it is desirable that any imaging system have a single *center of projection*. Strong cases in favor of a single viewpoint have also been made by Yamazawa et al. [Yamazawa *et al.*, 1995] and Nalwa [Nalwa, 1996]. Consider an image acquired by a sensor that can view the world in all directions from a single effective pinhole (see Figure 1). From such an omnidirectional image, pure perspective images can be constructed by mapping sensed brightness values onto a plane placed at any distance (effective focal length) from the viewpoint, as shown in Figure 1. Any image computed in this manner preserves linear perspective geometry. Images that adhere to perspective projection are desirable from two standpoints; they are consistent with the way we are used to seeing images, and they lend themselves to further processing by the large body of work in computational vision that assumes linear perspective projection.

*This work was supported in parts by the DARPA/ONR MURI Grant N00014-95-1-0601, an NSF National Young Investigator Award, and a David and Lucile Packard Fellowship.

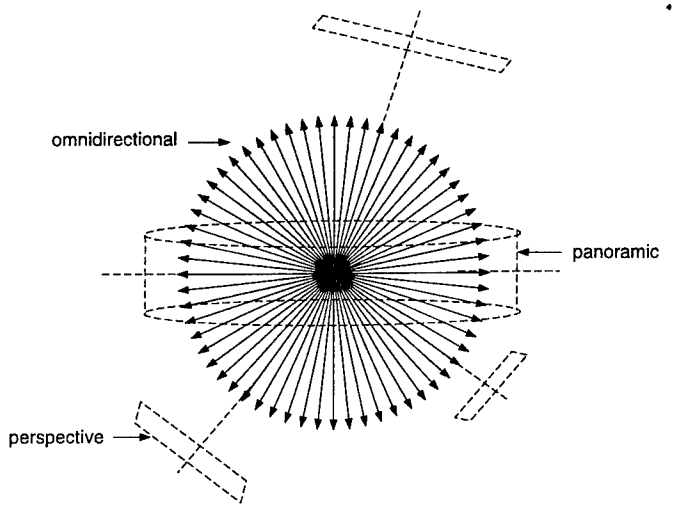


Figure 1: A truly omnidirectional image sensor views the world through an entire "sphere of view" as seen from its center of projection. The single viewpoint permits the construction of pure perspective images (computed by planar projection) or a panoramic image (computed by cylindrical projection). Panoramic sensors are not equivalent to omnidirectional sensors as they are omnidirectional only in one of the two angular dimensions.

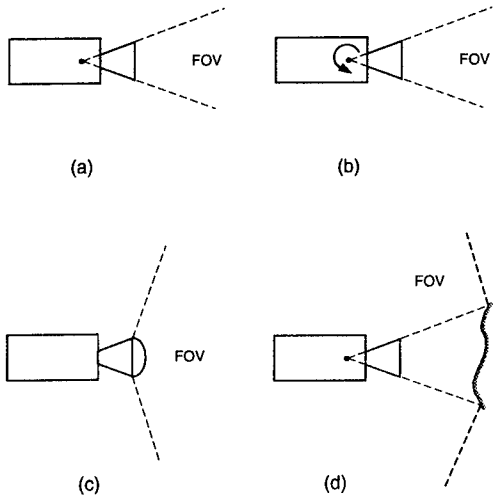


Figure 2: (a) A conventional imaging system and its limited field of view. A larger field of view may be obtained by (b) rotating the imaging system about its center of projection, (c) appending a fish-eye lens to the imaging system, and (d) imaging the scene through a mirror.

3 State of the Art

Before we present our omnidirectional camera, a review of existing imaging systems that seek to achieve wide fields of view is in order. An excellent review of some of the previous work can be found in [Nalwa, 1996].

3.1 Traditional Imaging Systems

Most imaging systems in use today comprise of a video camera, or a photographic film camera, attached to a lens. The image projection model for most camera lenses is perspective with a single center of projection. Since the imaging device (CCD array, for instance) is of finite size and the camera lens occludes itself while receiving incoming rays, the lens typically has a small field of view that corresponds to a small cone rather than a hemisphere (see Figure 2(a)). At first thought, it may appear that a large field can be sensed by packing together a number of cameras, each one pointing in a different direction. However, since the centers of projection reside inside their respective lenses, such a configuration proves infeasible.

3.2 Rotating Imaging Systems

An obvious solution is to rotate the entire imaging system about its center of projection, as shown in Figure 2(b). The sequence of images acquired by rotation are "stitched" together to obtain a panoramic view of the scene. Such an approach has been recently proposed by several investigators (see [Chen, 1995], [McMillan and Bishop, 1995], [Krishnan and Ahuja, 1996], [Zheng and Tsuji, 1990]). Of these the most novel is the system developed by Krishnan and Ahuja [Krishnan and Ahuja, 1996] which uses a camera with a non-frontal image detector to scan the world.

The first disadvantage of any rotating imaging system is that it requires the use of moving parts and precise positioning. A more serious drawback lies in the total time required to obtain an image with enhanced field of view. This restricts the use of rotating systems to static scenes and non-real-time applications.

3.3 Fish-Eye Lenses

An interesting approach to wide-angle imaging is based on the fish-eye lens (see [Wood, 1906], [Miyamoto, 1964]). Such a lens is used in place of a conventional camera lens and has a very short focal length that enables the camera to view objects within as much as a hemisphere (see Figure 2(c)). The use of fish-eye lenses for wide-angle imaging has been advocated in [Oh and Hall, 1987] and [Kuban *et al.*, 1994], among others.

It turns out that it is difficult to design a fish-eye lens that ensures that all incoming principal rays intersect at a single point to yield a fixed viewpoint (see [Nalwa, 1996] for details). This is indeed a problem with commercial fish-eye

lenses, including, Nikon's Fisheye-Nikkor 8mm f/2.8 lens. In short, the acquired image does not permit the construction of distortion-free perspective images of the viewed scene (though constructed images may prove good enough for some visualization applications). In addition, to capture a hemispherical view, the fish-eye lens must be quite complex and large, and hence expensive.

3.4 Catadioptric Systems

As shown in Figure 2(d), a catadioptric imaging system uses a reflecting surface to enhance the field of view. The rear-view mirror in a car is used exactly in this fashion. However, the shape, position, and orientation of the reflecting surface are related to the viewpoint and field of view in a complex manner. While it is easy to construct a configuration which includes one or more mirrors that dramatically increase the field of view of the imaging system, it is hard to keep the effective viewpoint fixed in space. Examples of catadioptric image sensors can be found in [Yagi and Kawato, 1990], [Hong, 1991], [Yamazawa *et al.*, 1995], and [Nalwa, 1996]. A recent theoretical result (see [Nayar and Baker, 1997]) reveals the complete class of catadioptric imaging systems that satisfy the single viewpoint constraint. This general solution has enabled us to evaluate the merits and drawbacks of previous implementations as well as suggest new ones [Nayar and Baker, 1997].

Here, we will briefly summarize previous approaches. In [Yagi and Kawato, 1990], a conical mirror is used in conjunction with a perspective lens. Though this provides a panoramic view, the single viewpoint constraint is not satisfied. The result is a viewpoint locus that hangs like a halo over the mirror. In [Hong, 1991], a spherical mirror was used with a perspective lens. Again, the result is a large locus of viewpoints rather than a single point. In [Yamazawa *et al.*, 1995], a hyperboloidal mirror used with a perspective lens is shown to satisfy the single viewpoint constraint. This solution is a useful one. However, the sensor must be implemented and calibrated with care. More recently, in [Nalwa, 1996], a novel panoramic sensor has been proposed that includes four planar mirrors that form the faces of a pyramid. Four separate imaging systems are used, each one placed above one of the faces of the pyramid. The optical axes of the imaging systems and the angles made by the four planar faces are adjusted so that the four viewpoints produced by the planar mirrors coincide. The result is a sensor that has a single viewpoint and a panoramic field of view of approximately $360° \times 50°$. Again, careful alignment and calibration are needed during implementation.

4 Omnidirectional Camera

While all of the above approaches use mirrors placed in the view of perspective lenses, we approach the problem using an orthographic lens. It is easy to see that if image projection is orthographic rather than perspective, the geometrical mappings between the image, the mirror and the world are invariant to translations of the mirror with respect to the imaging system. Consequently, both calibration as well as the computation of perspective images is greatly simplified.

There are several ways to achieve orthographic projection, of which we shall mention a few. The most obvious of these is to use commercially available telecentric lenses [Edmund Scientific, 1996] that are designed to be orthographic. It has also been shown [Watanabe and Nayar, 1996] that precise orthography can be achieved by simply placing an aperture [Kingslake, 1983] at the back focal plane of an off-the-shelf lens. Further, several zoom lenses can be adjusted to produce orthographic projection. Yet another approach is to mount an inexpensive relay lens onto an off-the-shelf perspective lens. The relay lens not only converts the imaging system to an orthographic one but can also be used to undo more subtle optical effects such as coma and astigmatism [Born and Wolf, 1965] produced by curved mirrors. In short, the implementation of pure orthographic projection is viable and easy to implement.

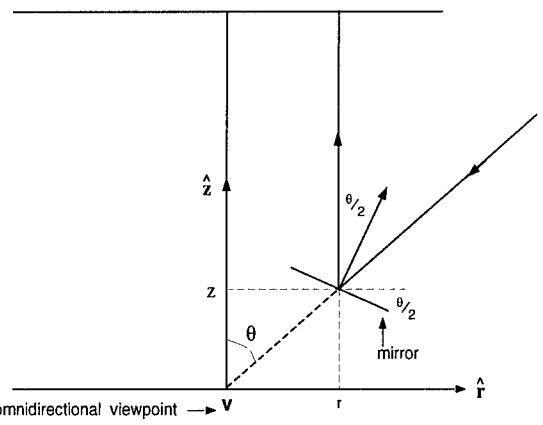


Figure 3: Geometry used to derive the reflecting surface that produces an image of the world as seen from a fixed viewpoint $\mathbf{v}$. This image is captured using an orthographic (telecentric) imaging lens.

We are now ready to derive the shape of the reflecting surface. Since orthographic projection is rotationally symmetric, all we need to determine is the cross-section $z(r)$ of the reflecting surface. The mirror is then the solid of revolution obtained by sweeping the cross-section about the

axis of orthographic projection. As illustrated in Figure 3, each ray of light from the world heading in the direction of the viewpoint $\mathbf{v}$ must be reflected by the mirror in the direction of orthographic projection. The relation between the angle θ of the incoming ray and the profile $z(r)$ of the reflecting surface is

$$\tan\theta = \frac{r}{z}. \quad (1)$$

Since the surface is specular, the angles of incidence and reflectance are equal to $\theta/2$. Hence, the slope at the point of reflection can be expressed as

$$\frac{dz}{dr} = -\tan\frac{\theta}{2}. \quad (2)$$

Now, we use the trignometric identity

$$\tan\theta = \frac{2\tan\frac{\theta}{2}}{1-\tan^2\frac{\theta}{2}}. \quad (3)$$

Substituting (1) and (2) in the above expression, we obtain

$$\frac{-2\frac{dz}{dr}}{1-(\frac{dz}{dr})^2} = \frac{r}{z}. \quad (4)$$

Thus, we find that the reflecting surface must satisfy a quadratic first-order differential equation. The first step is to solve the quadratic expression for surface slope. This gives us two solutions of which only one is valid since the slope of the surface in the first quadrant is assumed to be negative (see Figure 3):

$$\frac{dz}{dr} = \frac{z}{r} - \sqrt{1+(\frac{r}{z})^2}. \quad (5)$$

This first-order differential equation can be solved to obtain the following expression for the reflecting surface:

$$z = \frac{h^2 - r^2}{2h}, \quad (6)$$

where, $h > 0$ is the constant of integration.

Not surprisingly, the mirror that guarantees a single viewpoint for orthographic projection is a paraboloid. Paraboloidal mirrors are frequently used to converge an incoming set of parallel rays at a single point (the focus), or to generate a collimated light source from a point source (placed at the focus). In both these cases, the paraboloid is a concave mirror that is reflective on its inner surface. In our case, the paraboloid is reflective on its outer surface (convex mirror); all incoming principle rays are orthographically reflected by the mirror but can be extended to intersect at its focus, which serves as the viewpoint. Note that a concave paraboloidal mirror can also be used (this corresponds to the second solution we would get from equation (4) if the slope of the mirror in the first quadrant is assumed to be positive). This solution is less desirable to us since incoming rays with large angles of incidence θ would be self-occluded by the mirror.

A shown in Figure 4, parameter h of the paraboloid is its radius at $z = 0$. The distance between the vertex and the focus is $h/2$. Therefore, h determines the size of the paraboloid that, for any given orthographic lens system, can be chosen to maximize resolution. Shortly, the issue of resolution will be addressed in more detail.

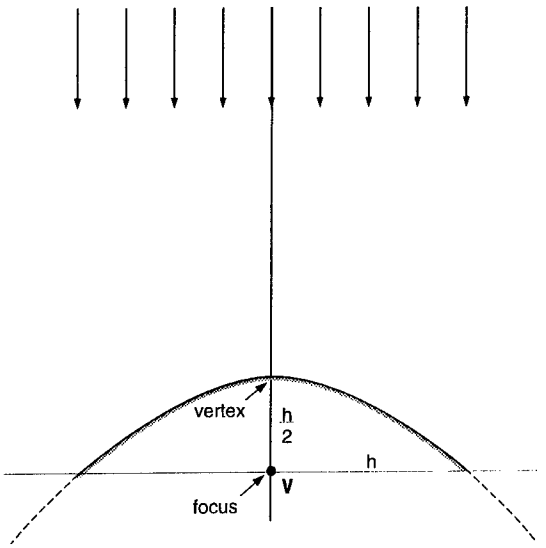


Figure 4: For orthographic projection, the solution is a paraboloid with the viewpoint located at the focus. Orthographic projection makes the geometric mappings between the image, the paraboloidal mirror and the world invariant to translations of the mirror. This greatly simplifies calibration and the computation of perspective images from paraboloidal ones.

5 Field of View

As the extent of the paraboloid increases, so does the field of view of the catadioptric sensor. It is not possible, however, to acquire the entire sphere of view since the paraboloid itself must occlude the world beneath it. This brings us to an interesting practical consideration: Where should the paraboloid be terminated? Note that

$$\left|\frac{dz}{dr}\right|_{z=0} = 1. \quad (7)$$

Hence, if we cut the paraboloid at the plane $z = 0$, the field of view exactly equals the upper hemisphere (minus the solid angle subtended by the imaging system itself). If a field of view greater

than a hemisphere is desired, the paraboloid can be terminated below the $z = 0$ plane. If only a panorama is of interest, an annular section of the paraboloid may be obtained by truncating it below and above the $z = 0$ plane. For that matter, given any desired field of view, the corresponding section of the parabola can be used and the entire resolution of the imaging device can be dedicated to that section's projection in the image.

In our prototypes, we have chosen to terminate the parabola at the $z = 0$ plane. This proves advantageous in applications in which the complete sphere of view is desired, as shown in Figure 5. Since the paraboloid is terminted at the focus, it is possible to place two identical catadioptric cameras back-to-back such that their foci (viewpoints) coincide. Thus, we have a truly omnidirectional sensor, one that is capable of acquiring an entire sphere of view at video rate.

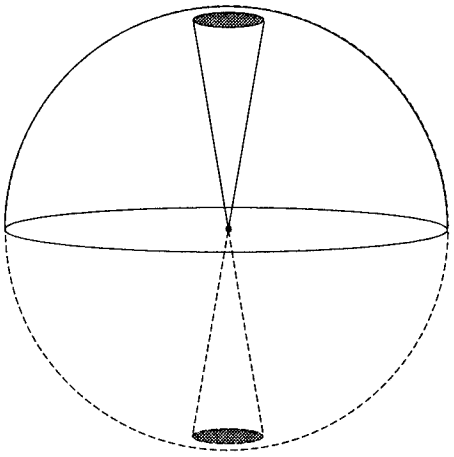
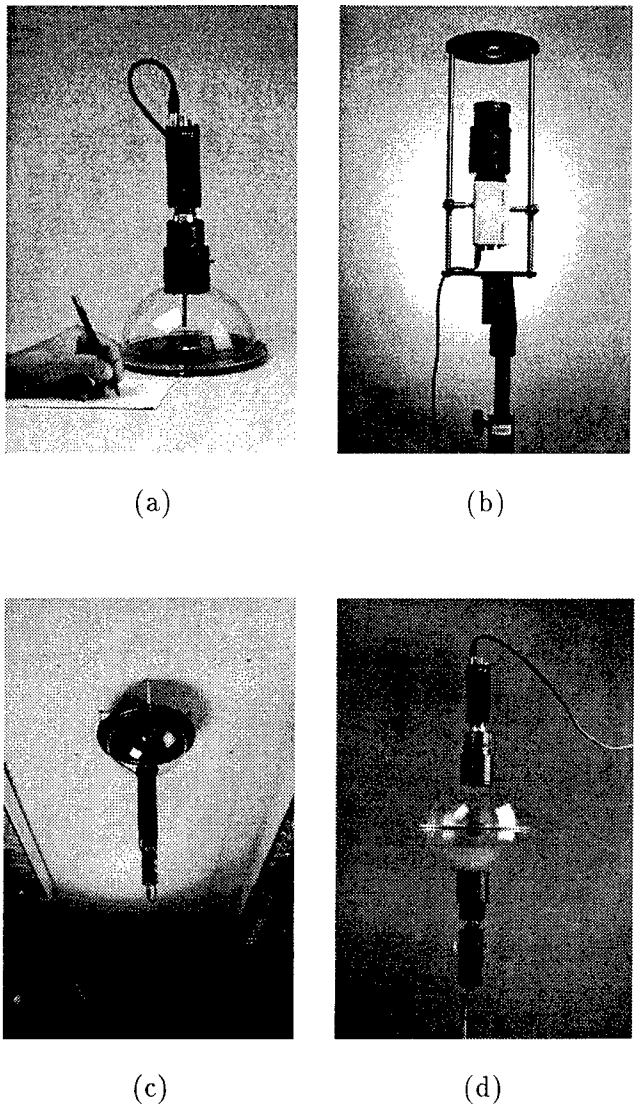


Figure 5: If the paraboloid is cut by the horizontal plane that passes through its focus, the field of view of the catadioptric system exactly equals the upper hemisphere. This allows us to place two catadioptric sensors back-to-back such that their foci (viewpoints) coincide. The result is a truly omnidirectional sensor that can acquire the entire sphere of view. The shaded regions are parts of the field of view where the sensor sees itself.



(a) (b) (c) (d)

Figure 6: Four implementations of catadioptric omnidirectional video cameras that use paraboloidal mirrors. (a) This compact sensor for *teleconferencing* uses a 1.1 inch diameter paraboloidal mirror, a Panasonic GP-KR222 color camera, and Cosmicar/Pentax C6Z1218 zoom and close-up lenses to achieve orthography. The transparent spherical dome minimizes self-obstruction of the field of view. (b) This camera for *navigation* uses a 2.2 inch diameter mirror, a DXC-950 Sony color camera, and a Fujinon CVL-713 zoom lens. The base plate has an attachment that facilitates easy mounting on mobile platforms. (c) This sensor for *surveillance* uses a 1.6 inch diameter mirror, an Edmund Scientific 55mm F/2.8 telecentric (orthographic) lens and a Sony XR-77 black and white camera. The sensor is lightweight and suitable for mounting on ceilings and walls. (d) This sensor is a back-to-back configuration that enables it to sense the entire sphere of view. Each of its two units is identical to the sensor in (a).

6 Implementation

Several versions of the proposed omnidirectional sensor have been built, each one geared towards a specific application. The applications we have in mind include video teleconferencing, remote surveillance and autonomous navigation. Figure 6 shows and details the different sensors and their components. The basic components of all the sensors are the same; each one includes a paraboloidal mirror, an orthographic lens system and a CCD video camera. The sensors differ primarily in the their mechanical designs and their attachments. For instance, the sensors in

Figures 6(a) and 6(c) have transparent spherical domes that minimize self-obstruction of their hemispherical fields of view. Figure 6(d) shows a back-to-back implementation that is capable of acquiring the complete sphere of view.

The use of paraboloidal mirrors virtually obviates calibration. All that is needed are the image coordinates of the center of the paraboloid and its radius h. Both these quantities are measured in pixels from a single omnidirectional image. We have implemented software for the generation of perspective images. First, the user specifies the viewing direction, the image size and effective focal length (zoom) of the desired perspective image (see Figure 1). Again, all these quantities are specified in pixels. For each three-dimensional pixel location (x_p,y_p,z_p) on the desired perspective image plane, its line of sight with respect to the viewpoint is computed in terms of its polar and azimuthal angles:

$$\theta = \cos^{-1} \frac{z_p}{\sqrt{{x_p}^2 + {y_p}^2 + {z_p}^2}}, \quad \phi = \tan^{-1} \frac{y_p}{x_p}. \tag{8}$$

This line of sight intersects the paraboloid at a distance ρ from its focus (origin), which is computed using the following spherical expression for the paraboloid:

$$\rho = \frac{h}{(1 + \cos\theta)}. \tag{9}$$

The brightness (or color) at the perspective image point (x_p,y_p,z_p) is then the same as that at the omnidirectional image point

$$x_i = \rho \sin\theta \cos\phi, \quad y_i = \rho \sin\theta \sin\phi. \tag{10}$$

The above computation is repeated for all points in the desired perspective image. Figure 7 shows an omnidirectional image (512x480 pixels) and several perspective images (200x200 pixels each) computed from it. It is worth noting that perspective projection is indeed preserved. For instance, straight lines in the scene map to straight lines in the perspective images while they appear as curved lines in the omnidirectional image. Recently, a video-rate version of the above described image generation has been developed as an interactive software system called OmniVideo [Peri and Nayar, 1997].

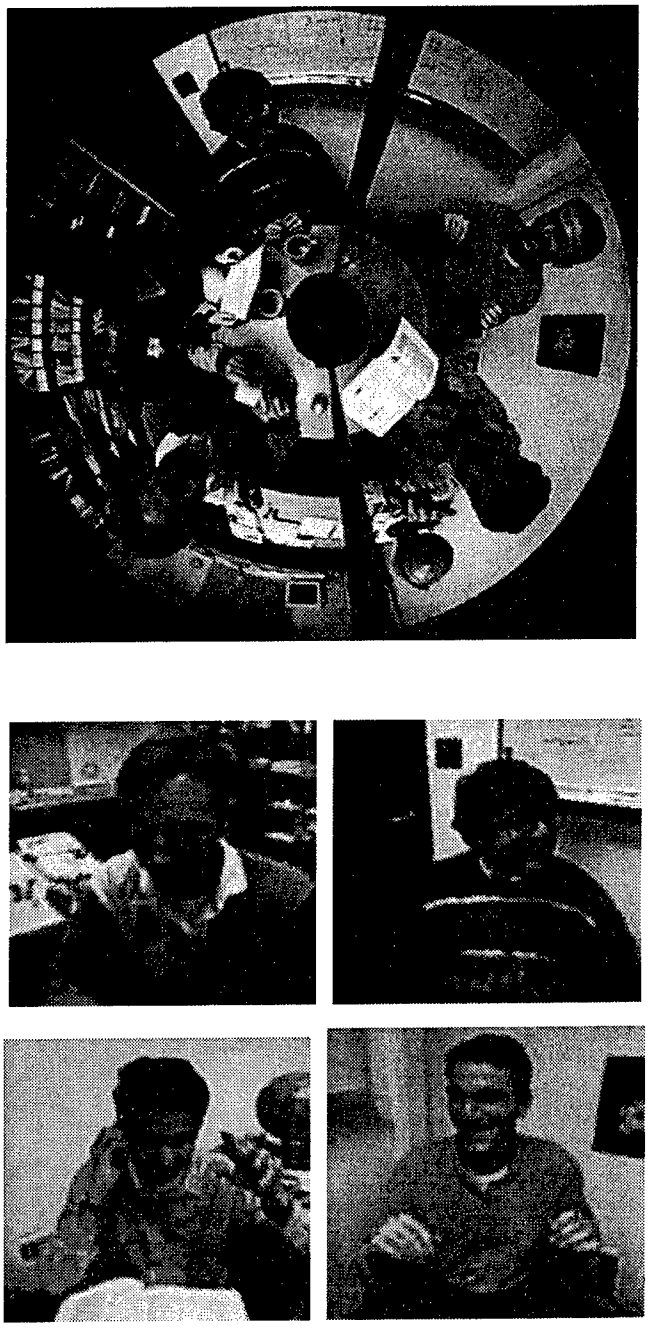


Figure 7: Software generation of perspective images (bottom) from an omnidirectional image (top). Each perspective image is generated using user-selected parameters, including, viewing direction (line of sight from the viewpoint to the center of the desired image), effective focal length (distance of the perspective image plane from the viewpoint of the sensor), and image size (number of desired pixels in each of the two dimensions). It is clear that the computed images are indeed perspective; for instance, straight lines are seen to appear as straight lines though they appear as curved lines in the omnidirectional image.

7 Resolution

Several factors govern the resolution of a catadioptric sensor. Let us begin with the most obvious of these, the spatial resolution due to finite pixel size. In [Nayar and Baker, 1997], we have derived a general expression for the spatial resolution of any catadioptric camera. In the case of

our paraboloidal mirror, the resolution increases by a factor of 4 from the vertex ($r = 0$) of the paraboloid to the fringe ($r = h$). In practice, this drop in resolution towards the center of the paraboloidal image is not easily discernible. In principle, it is of course possible to use image detectors with non-uniform resolution to compensate for the above variation. It should also be mentioned that while all our implementations use CCD arrays with 512x480 pixels, nothing precludes us from using detectors with 1024x1024 or 2048x2048 pixels that are commercially available at a higher cost.

More intriguing are the blurring effects of coma and astigmatism that arise due to the aspherical nature of the reflecting surface [Born and Wolf, 1965]. Since these effects are linear but shift-variant [Robbins and Huang, 1972], a suitable set of deblurring filters need to be explored. Alternatively, these effects can be significantly reduced using inexpensive corrective lenses.

Acknowledgements

This work was inspired by the prior work of Vic Nalwa of Lucent Technologies. I have benefitted greatly from discussions with him. I thank Simon Baker and Venkata Peri of Columbia University for their valuable comments on various drafts of this paper.

References

[Born and Wolf, 1965] M. Born and E. Wolf. *Principles of Optics.* London:Permagon, 1965.

[Chen, 1995] S. E. Chen. QuickTime VR - An Image Based Approach to Virtual Environment Navigation. *Computer Graphics: Proc. of SIGGRAPH 95*, pages 29–38, August 1995.

[Edmund Scientific, 1996] *1996 Optics and Optical Components Catalog*, volume 16N1. Edmund Scientific Company, New Jersey, 1996.

[Goshtasby and Gruver, 1993] A. Goshtasby and W. A. Gruver. Design of a Single-Lens Stereo Camera System. *Pattern Recognition*, 26(6):923–937, 1993.

[Hong, 1991] J. Hong. Image Based Homing. *Proc. of IEEE International Conference on Robotics and Automation*, May 1991.

[Kingslake, 1983] R. Kingslake. *Optical System Design.* Academic Press, 1983.

[Krishnan and Ahuja, 1996] A. Krishnan and N. Ahuja. Panoramic Image Acquisition. *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR-96)*, pages 379–384, June 1996.

[Kuban *et al.*, 1994] D. P. Kuban, H. L. Martin, S. D. Zimmermann, and N. Busico. Omniview Motionless Camera Surveillance System. *United States Patent No. 5,359,363*, October 1994.

[McMillan and Bishop, 1995] L. McMillan and G. Bishop. Plenoptic Modeling: An Image-Based Rendering System. *Computer Graphics: Proc. of SIGGRAPH 95*, pages 39–46, August 1995.

[Miyamoto, 1964] K. Miyamoto. Fish Eye Lens. *Journal of Optical Society of America*, 54(8):1060–1061, August 1964.

[Nalwa, 1996] V. Nalwa. A True Omnidirectional Viewer. Technical report, Bell Laboratories, Holmdel, NJ 07733, U.S.A., February 1996.

[Nayar and Baker, 1997] S. K. Nayar and S. Baker. Catadioptric Image Formation. *Proc. of DARPA Image Understanding Workshop*, May 1997.

[Nayar, 1988] S. K. Nayar. Sphereo: Recovering depth using a single camera and two specular spheres. *Proc. of SPIE: Optics, Illumumination, and Image Sensing for Machine Vision II*, November 1988.

[Oh and Hall, 1987] S. J. Oh and E. L. Hall. Guidance of a Mobile Robot using an Omnidirectional Vision Navigation System. *Proc. of the Society of Photo-Optical Instrumentation Engineers, SPIE*, 852:288–300, November 1987.

[Peri and Nayar, 1997] V. Peri and S. K. Nayar. Generation of Perspective and Panoramic Video from Omnidirectional Video. *Proc. of DARPA Image Understanding Workshop*, May 1997.

[Robbins and Huang, 1972] G. M. Robbins and T. S. Huang. Inverse Filtering for Linear Shift-Variant Imaging Systems. *Proceedings of the IEEE*, 60(7):862–872, July 1972.

[Watanabe and Nayar, 1996] M. Watanabe and S. K. Nayar. Telecentric optics for computational vision. *Proc. of European Conference on Computer Vision*, April 1996.

[Wood, 1906] R. W. Wood. Fish-eye views, and vision under water. *Philosophical Magazine*, 12(Series 6):159–162, 1906.

[Yagi and Kawato, 1990] Y. Yagi and S. Kawato. Panoramic Scene Analysis with Conic Projection. *Proc. of International Conference on Robots and Systems (IROS)*, 1990.

[Yamazawa *et al.*, 1995] K. Yamazawa, Y. Yagi, and M. Yachida. Obstacle Avoidance with Omnidirectional Image Sensor HyperOmni Vision. *Proc. of IEEE International Conference on Robotics and Automation*, pages 1062–1067, May 1995.

[Zheng and Tsuji, 1990] J. Y. Zheng and S. Tsuji. Panoramic Representation of Scenes for Route Understanding. *Proc. of the Tenth International Conference on Pattern Recognition*, 1:161–167, June 1990.

Generation of Perspective and Panoramic Video from Omnidirectional Video *

Venkata N. Peri and Shree K. Nayar
Department of Computer Science, Columbia University
New York, New York 10027
Email: {venkat, nayar}@cs.columbia.edu

Abstract

Existing software systems for visual exploration are limited in their capabilities in that they are only applicable to static omnidirectional images. We present a software system that has the capability to generate at video rate (30 Hz), a large number of perspective and panoramic video streams from a single omnidirectional video input, using no more than a PC. This permits a remote user to create multiple perspective and panoramic views of a dynamic scene, where the parameters of each view (viewing direction, field of view, and magnification) are controlled via an interactive device such as a mouse, joystick or a head-tracker.

1 Introduction

Remote visual exploration systems such as QuickTime® VR [Chen-1995] allow a user to navigate around a visual environment. This is done by simulating a virtual camera whose parameters are controlled by the user. A fundamental limitation of existing systems is that they are restricted to static environments, i.e. a single wide-angle image of a scene. The static image is typically obtained by stitching together several images of a static scene taken by rotating a camera about its center of projection. Only recently, it has become possible to acquire omnidirectional images at video rate (see [Nayar-1997]). The availability of such an acquisition device opens up the possibility of a software system that can create perspective and panoramic video streams. This adds a new dimension to the notion of remote visual exploration.

The omnidirectional camera developed by Nayar captures at video rate, a hemispherical field of view as seen from a single point. We have developed a real-time software system called OmniVideo that can generate multiple perspective and panoramic video streams from such an omnidirectional video stream. The user can create and orient multiple perspective and panoramic views in desired directions; all views are updated at video rate. Furthermore, the viewing direction, field of view, and magnification of each video stream can be controlled using an interactive device such as a mouse, joystick, or a head-tracker. The capabilities of the OmniVideo system can be exploited in a variety of applications, including immersive video, teleconferencing, autonomous navigation, and video surveillance and monitoring. We have also developed an omnidirectional web-camera wherein, view parameters can be modified using a control panel on the client's browser. An online demonstration is available at http://omnicam.cs.columbia.edu/.

2 The OmniVideo System

In the OmniVideo system, the omnidirectional video input defines the dynamic visual environment. Perspective and panoramic views are essentially virtual cameras positioned in this visual environment. Navigation and exploration of this visual environment is performed by modifying one or more camera parameters. Perspective and panoramic virtual cameras have five parameters, namely pan, tilt, zoom, roll, and field of view. In the OmniVideo system, the user can modify these parameters using an interactive device such as a mouse, joystick, or a head-tracker.

2.1 Reprojection

The optics of the omnidirectional camera is designed to reflect a wide-angle view orthographically off a parabolic mirror, onto the sensing element (CCD) of a conventional camera (see [Nayar-1997]). Views are generated by computing pixel intensities of every pixel $P(x_p, y_p, z_p)$ on the imaging surface of the virtual camera. Pixel in-

* This work was supported in parts by the DARPA/ONR MURI Grant N00014-95-1-0601, an NSF National Young Investigator Award, and a David and Lucile Packard Fellowship.

tensities are determined by reprojection. This is equivalent to determining the intensity of the point of intersection of the ray $R(\theta,\phi)$ (θ and ϕ are polar and azimuthal angles, respectively) from the focus of the parabolic mirror, in the direction of the point P.

The equation of the parabolic mirror is given by

$$z(r)=\frac{h^2-r^2}{2h},\quad r^2=x^2+y^2,\quad h>0, \qquad (1)$$

where h is the *parameter* of the parabola. The ray $R(\theta,\phi)$ intersects the parabola at a distance

$$\rho=h/(1+\cos\theta) \qquad (2)$$

from the focus. When projected orthographically on to the CCD, the coordinates of the point of intersection are given by

$$x_i=\rho\sin\theta\cos\phi,\quad y_i=\rho\sin\theta\sin\phi. \qquad (3)$$

Interpolation is used to determine intensity at this point.

Rewriting equation (3) we get

$$x_i=\frac{h}{z_p+\sqrt{x_p^2+y_p^2+z_p^2}}x_p, \qquad (4)$$

$$y_i=\frac{h}{z_p+\sqrt{x_p^2+y_p^2+z_p^2}}y_p. \qquad (5)$$

This form of equation (3) is suitable for optimization, as we shall see later.

2.2 Implementation

Video-rate performance is the most important feature of the OmniVideo system. Since the incoming visual information changes dynamically, OmniVideo *cannot* take advantage of most real-time reprojection methods that have been developed for static images (see [Chen-1995], [McMillan and Bishop-1995], [Lippman-1980], [Miller and Chen-1993]). We have implemented several numerical and data optimizations, which give to video-rate performance on a PC.

We use the notion of *geometric maps* to generate views. The geometric map defines the coordinate transformation between pixels on the imaging surface of the perspective (or panoramic) virtual camera, and the omnidirectional image. Simply stated, the geometric map implements the coordinate transformations of equations (4) and (5) in a lookup table. The process of reprojection is reduced to a lookup through the geometric map. The geometric map of a view changes only when viewing parameters of the associated virtual camera change. In computing views at video-rate, we observe that geometric maps provide the greatest speedup when viewing parameters are unmodified.

Yet another speedup is the use of lookup tables for geometric map generation. In equations (4) and (5), the term $\sqrt{x_p^2+y_p^2+z_p^2}$ represents the *distance* of the pixel from the focus of the parabola. As we shall see, this *distance* factor can be rewritten in a manner that is independent of all view parameters, except zoom, and hence can be determined from a lookup table.

For a perspective view, this *distance* can be written as $\sqrt{x_v^2+y_v^2+f^2}$, where (x_v,y_v) are the coordinates of the pixel $P(x_p,y_p,z_p)$ in the coordinate system of the virtual camera. f is the focal length of the perspective view. Substituting, equations (4) and (5) can be written as:

$$x_i=\frac{h}{z_p+\sqrt{x_v^2+y_v^2+f^2}}x_p, \qquad (6)$$

$$y_i=\frac{h}{z_p+\sqrt{x_v^2+y_v^2+f^2}}y_p. \qquad (7)$$

Similarly, for a panoramic view, the *distance* factor $\sqrt{x_p^2+y_p^2+z_p^2}$ can be expressed as $\sqrt{w^2+f^2}$, where w is the *height* of the pixel along the cylindrical surface of projection, and f is the focal length (radius of the cylinder). Again, equations (4) and (5) become:

$$x_i=\frac{h}{z_p+\sqrt{w^2+f^2}}x_p, \qquad (8)$$

$$y_i=\frac{h}{z_p+\sqrt{w^2+f^2}}y_p. \qquad (9)$$

The *distance* factor now is effectively a constant. Using the optimized equations for x_i and y_i (equations (6), (7), (8) and (9)), it is possible to compute geometric maps of perspective and panoramic views at video rate.

Since reprojection of pixel coordinates (x_p,y_p,z_p) takes place in a raster scan manner, there is tremendous computational redundancy in inter-pixel computations. We exploit this redundancy by incremental computation of pixel coordinates.

A critical implementation issue, especially in video applications, is the overlap between computation, user interaction, and video display. OmniVideo takes advantage of the multithreading available in most modern operating systems (such as Windows® NT) to provide a responsive user interface, while operating at full video-rate.

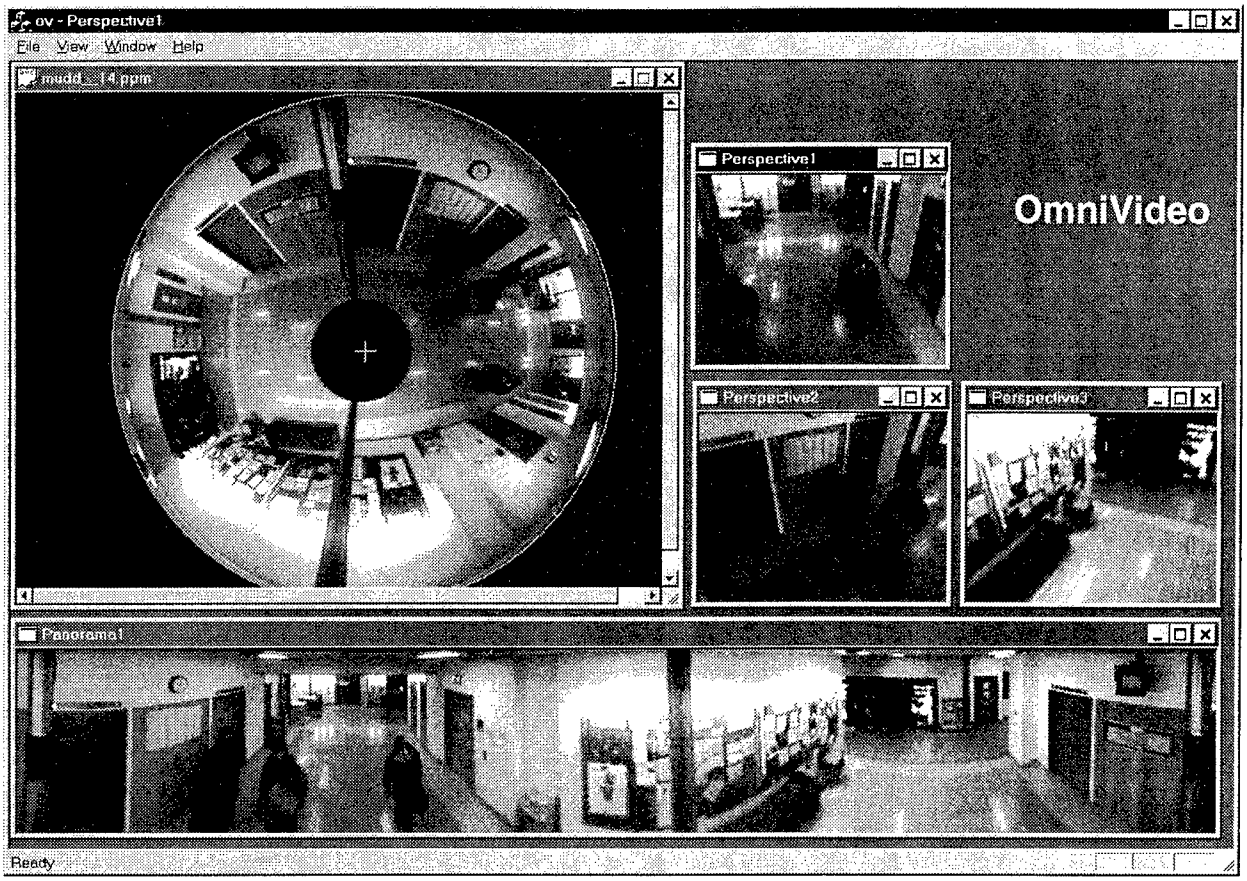


Figure 1: The OmniVideo system allows a user to generate multiple perspective and panoramic video streams from an incoming omnidirectional stream (top-left).

3 Results

We have implemented OmniVideo on an IBM compatible Pentium Pro PC, operating at 200 MHz. The system has a simple interface that allows the user to control viewing parameters using either a joystick or a mouse. In this configuration, OmniVideo can generate up to 12 perspective and panoramic video streams at video rate. Figure 1 shows the system in a typical surveillance and monitoring application.

A novel application for the OmniVideo system is an omnidirectional web-camera. The OmniVideo system is integrated with an http server (such as Microsoft's Internet Information Server). A live omnidirectional camera feeds into the OmniVideo system. Multiple users can connect to the web-camera and navigate the scene captured by the omnidirectional camera, in real time over the Internet. View parameters are controlled using a control panel provided on the client's browser. The server running on a 200 MHz Pentium Pro PC can support a large number of connections at video-rates. An online demonstration is available at http://omnicam.cs.columbia.edu/.

References

[Chen, 1995] S. E. Chen. QuickTime VR – An Image Based Approach to Virtual Environment Navigation. *Computer Graphics: Proc. Of SIGGRAPH 95,* pages 29-38, August 1995.

[Lippman, 1980] A. Lippman, Movie Maps: An Application of the Optical videodisk to Computer Graphics, *Proc. Of SIGGRAPH 80,* 1980.

[McMillan and Bishop, 1995] L. McMillan and G. Bishop. Plenoptic Modeling: An Image-Based Rendering System. *Computer Graphics: Proc. Of SIGGRAPH 95,* pages 39-46, August 1995.

[Miller and Chen, 1993] G. Miller and S. E. Chen. Real-Time Display of Surroundings using Environment Maps. *Technical Report No. 44, Apple Computer, Inc.,* 1993.

[Nayar, 1997] S. K. Nayar. Omnidirectional Video Camera. *Proc. Of DARPA Image Understanding Workshop,* May 1997.

An Integrated Approach to Image Stabilization, Mosaicking and Super-resolution

S. Srinivasan R. Chellappa
Center for Automation Research, University of Maryland
College Park, MD 20742-3275

Abstract

Image stabilization, mosaicking and super-resolution are fundamental image sequence operations that are linked by the common thread of image motion analysis. The accuracy of these processes is tied closely to the accuracy with which interframe motion is estimated for the sequence. In this paper, we formulate a unified framework for solving these problems based on a robust technique for computing optical flow using overlapped basis functions. We develop specific algorithms that robustly combine the flow estimates to give as their outputs the stabilized sequence, image mosaic and high-resolution image.

1 Introduction

An image sequence gathered by a remote camera (*e.g.* by a camera mounted on an unmanned air or ground vehicle) calls for significant pre-processing before it can be exploited by an automated algorithm or a tele-operator. Often the 3D motion of the camera platform and 3D structure of the imaged scene are not known to an accuracy that allows for characterizing the temporal evolution of the sequence. Such is the case when (*i*) the terrain over which the platform moves is unknown, (*ii*) the camera is uncalibrated or has lost its calibration, (*iii*) camera pan/zoom cannot be estimated reliably, or (*iv*) the scene being viewed has rich 3D structure. The first step in consolidating an image sequence is to compensate for "unwanted" camera motion by a process known as *image stabilization*. When operating in a remote environment, it is desirable to obtain as large a field of view as possible, if necessary by moving the platform or panning the camera. The process of piecing together the information in each frame of the sequence to build a representation encompassing a larger field of view is known as *mosaicking*. The temporal coherence in a video sequence implies data redundancy, and this allows for improving the image quality of the mosaic. This improvement results from suppressing the noise components of individual frames while building the mosaic, and enhancing the resolution of the denoised mosaic through a process known as *super-resolution*. These processes of image stabilization, mosaicking and super-resolution are linked by the common thread of *image motion analysis*.

In this paper, we present an integrated approach to these image motion analysis processes. Our system is built around the robust flow field technique described in [9]. An overview of the system architecture is presented in Fig.1. The key innovations proposed here include an improved optical flow formulation, which is solved by a fast algorithm. Global motion parameters are estimated robustly from the local optical flow field. The stabilization process comprises a stage for removing gross, temporally correlated motion followed by a stage that compensates for residual uncorrelated jitter. A mosaic is robustly obtained from the image sequence and warping parameters. Super-resolution is achieved essentially by applying a sharpening filter to the mosaic. In using optical flow rather than a feature point technique for image motion computation, the sensitive dependence of algorithm accuracy on the detection and tracking accuracy of features is avoided. The superiority of this technique over others [1; 4; 10] lies in its emphasis on robustness at every stage, leading to improved immunity to noise, computational speed, and the ability to integrate multiple functionalities into a single computational framework.

This paper is organized as follows: The optical flow algorithm is described in Section 2 and the stabiliza-

The support of the Defense Advanced Research Projects Agency and the Office of Naval Research under Grant N00014-95-1-0521 is gratefully acknowledged.

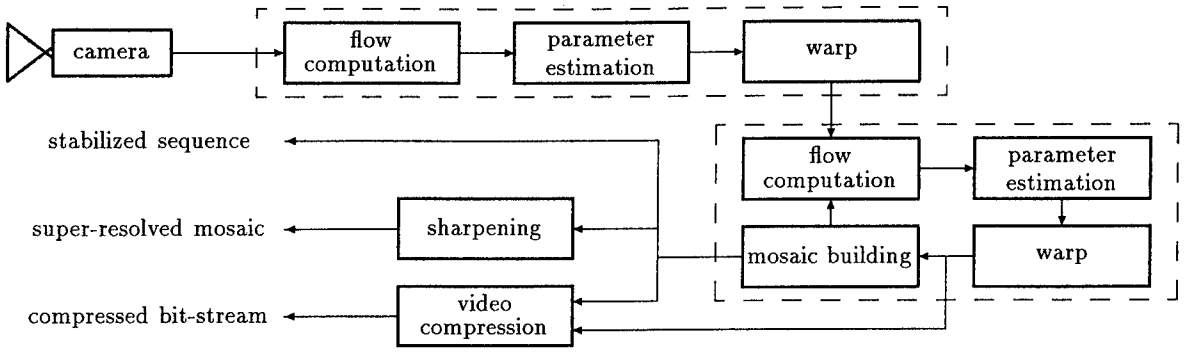


Figure 1: System architecture.

tion method in Section 3. Section 4 covers the temporal integration processes of mosaicking and super-resolution. Results of applying our algorithms to visual and IR imagery are discussed in Section 5.

2 Optical Flow Estimation

When the projected 2D image field of a scene is given by $\psi = \psi(x, y, t)$, preservation of luminance patterns implies the *gradient constraint equation*

$$\frac{\partial \psi}{\partial t} + u\frac{\partial \psi}{\partial x} + v\frac{\partial \psi}{\partial y} = 0 \qquad \forall x, y, t \qquad (1)$$

In (1), u and v denote the horizontal and vertical velocities (as functions of space and time) respectively. Together, they constitute the *optical flow* of the sequence. For every triplet (x, y, t) in (1), there are two unknowns, making the problem of computing the optical flow ill-conditioned. In practice, the system is regularized by imposing additional smoothness constraints on u and v. The performance of current solutions to (1) is often not consistent with respect to the accuracy and reliability of the field, and is characterized by numerical sensitivity, besides being computationally expensive.

2.1 Formulation

An alternative to computing the optical flow on a pixelwise basis is to model the motion fields u and v in terms of a weighted sum of basis functions and estimating the weights which constitute the model parameters. In this approach, the motion field is force-fitted to a local model and derives its smoothness properties from those of the model basis functions. Let $\{\phi = \phi(x, y, t)\}$ be a family of basis functions, and let the flow field be modeled as

$$u = \sum_{k=0}^{K} u_k \phi_k \quad \text{and} \quad v = \sum_{k=0}^{K} v_k \phi_k \qquad (2)$$

Substituting (2) into (1), we get

$$\frac{\partial \psi}{\partial t} + \sum_k u_k \phi_k \frac{\partial \psi}{\partial x} + \sum_k v_k \phi_k \frac{\partial \psi}{\partial y} = 0 \, \forall x, y, t \qquad (3)$$

This is a continuum of equations in 3-space which is reduced to a scalar equation for each instant of time by integrating with a multiplicative kernel $\theta = \theta(x, y)$:

$$\int \frac{\partial \psi}{\partial t}\theta dx dy + \sum_k u_k \int \phi_k \frac{\partial \psi}{\partial x}\theta dx dy + \sum_k v_k \int \phi_k \frac{\partial \psi}{\partial y}\theta dx dy = 0 \qquad (4)$$

Such an equation exists for every square-integrable kernel θ. In order to solve for $\{u_k, v_k\}$, it is necessary to choose appropriate kernels in (4).

System (3) is linear in the unknowns $\{u_k, v_k\}$ and is analogous to the matrix-vector system

$$Ax \rightarrow b \qquad A \in \Re^{M \times N}, M > N \qquad (5)$$

where x corresponds to the vector $(u_0, v_0, \ldots)'$. The analogy implies the applicability of solutions and results of (5) to (3). In the discrete domain, the analogy is obvious since an equation of type (3) exists for each pixel in the current frame, corresponding to one row of the composite matrix $[A|b]$. The least squares (LS) solution of (5) is given by $A'Ax = A'b$; choosing θ from the family $\{\phi_k \frac{\partial \psi}{\partial x}, \phi_k \frac{\partial \psi}{\partial y}\}$ gives the LS solution of (3). In practice, only discretized data is available for the image luminance field ψ. The LS solution assumes knowledge of the spatial derivatives of ψ, which may not be known reliably. Any minor and random non-compliance with (1) is accounted for in the observation error in b. A robust approach must try to minimize the sensitive dependence of the solution on spatial as well as temporal derivatives. In other words, in the analog (5), the solution must be accurate and robust to errors in A as well as in b. This requirement can be stated as follows:

Assume x_0 is the exact solution for the overconstrained linear system $Ax \rightarrow b$. Let Δ and δ be zero-mean, independent additive observation noise in A and b respectively, i.e. the quantities $\hat{A} = A + \Delta$ and $\hat{b} = b + \delta$ are observed. Find an optimal estimate x of x_0 given $\hat{A}$ and $\hat{b}$.

It can be shown that, under assumptions of uncorrelatedness of Δ and δ to A and b, and of invertibility

of the matrix $A'A$, neither the LS nor the total least squares (TLS) solutions of (5) are unbiased in the general case, and the corrected least squares (CLS) solution shows sensitive dependence on the errors in $\hat{A}$ as well as in $\hat{b}$ [9]. Interestingly, given the observations $\hat{G} = \hat{A}'A$ and $\hat{A}'\hat{b}$, the *Extended Least Squares* (ELS) solution (6) shows no dependence on the error Δ in the estimate of A:

$$\begin{aligned} x_E &= \hat{G}^{-1}\hat{A}\hat{b} \\ e_E &= \underbrace{\hat{G}^{-1}}_{\approx(A'A)^{-1}} [\Delta' \underbrace{(Ax_0 - b)}_{=0} + \underbrace{\Delta'\delta}_{O(2)} + A'\delta] \\ &\approx (A'A)^{-1}A'\delta \end{aligned} \tag{6}$$

Also, the observation corresponding to $\hat{G}$ is available for (3). In (6), x_E is unbiased, and has a smaller covariance than either x_{LS} or x_{CLS}. In the original problem (3), the ELS solution is obtained when θ in (4) is chosen from the family $\{\phi_k \frac{\partial\hat{\psi}}{\partial x}, \phi_k \frac{\partial\hat{\psi}}{\partial y}\}$ where the quantity $\frac{\partial\hat{\psi}}{\partial\{x,y\}}$ is an *estimate* of the derivative, giving the system

$$\begin{aligned} &\int \frac{\partial\hat{\psi}}{\partial t}\phi_l \frac{\partial\hat{\psi}}{\partial x}dxdy + \sum_k u_k \int \phi_k \frac{\partial\psi}{\partial x}\phi_l \frac{\partial\hat{\psi}}{\partial x}dxdy \\ &\quad + \sum_k v_k \int \phi_k \frac{\partial\psi}{\partial y}\phi_l \frac{\partial\hat{\psi}}{\partial x}dxdy = 0 \\ &\int \frac{\partial\hat{\psi}}{\partial t}\phi_l \frac{\partial\hat{\psi}}{\partial y}dxdy + \sum_k u_k \int \phi_k \frac{\partial\psi}{\partial x}\phi_l \frac{\partial\hat{\psi}}{\partial y}dxdy \\ &\quad + \sum_k v_k \int \phi_k \frac{\partial\psi}{\partial y}\phi_l \frac{\partial\hat{\psi}}{\partial y}dxdy = 0 \end{aligned} \tag{7}$$

with the estimated temporal derivative $\frac{\partial\hat{\psi}}{\partial t}$. The availability of the observations $\hat{G} = \hat{A}'A$ and $\hat{A}'\hat{b}$ is equivalent to the computability of the integrals in (7), under certain weak assumptions on the functional form of ϕ_k and the estimate $\frac{\partial\hat{\psi}}{\partial\{x,y\}}$.

Consider the integral $I(y) = \int \phi_k \frac{\partial\psi}{\partial x}\phi_l \frac{\partial\hat{\psi}}{\partial x}dx$. Assume that the estimate $\frac{\partial\hat{\psi}}{\partial\{x,y\}}$ has a differentiable functional form, *i.e.* the derivative $\frac{\partial}{\partial\{x,y\}}\frac{\partial\hat{\psi}}{\partial\{x,y\}}$ is known exactly. This holds for even the simplest discrete gradient masks like $(\ldots, 0, -1, 1, 0, \ldots)$ since the masks assume a smooth underlying functional form. Also, assume that $\{\phi_k\}$ are differentiable and that $\phi_k(x,y) \to 0$ as $x \to \pm\infty$ or $y \to \pm\infty$. Integrating by parts over $(-\infty, \infty)$, we obtain

$$I(y) = \underbrace{\left[\phi_k\phi_l\frac{\partial\hat{\psi}}{\partial x}\psi\right]}_{=0} - \int \psi\frac{\partial\phi_k\phi_l\frac{\partial\hat{\psi}}{\partial x}}{\partial x}dx \tag{8}$$

which is computable reliably without knowing the exact derivatives $\frac{\partial\psi}{\partial\{x,y\}}$. Applying this reasoning to (7) gives

$$\begin{aligned} \sum_k u_k \int \frac{\partial\phi_k\phi_l\frac{\partial\hat{\psi}}{\partial x}}{\partial x}\psi + \sum_k v_k \int \frac{\partial\phi_k\phi_l\frac{\partial\hat{\psi}}{\partial x}}{\partial y}\psi \\ = \int \frac{\partial\hat{\psi}}{\partial t}\phi_l\frac{\partial\hat{\psi}}{\partial x} \\ \sum_k u_k \int \frac{\partial\phi_k\phi_l\frac{\partial\hat{\psi}}{\partial y}}{\partial x}\psi + \sum_k v_k \int \frac{\partial\phi_k\phi_l\frac{\partial\hat{\psi}}{\partial y}}{\partial y}\psi \\ = \int \frac{\partial\hat{\psi}}{\partial t}\phi_l\frac{\partial\hat{\psi}}{\partial y} \end{aligned} \tag{9}$$

where the integrals are over the entire X-Y plane and can be computed reliably. (9) has the following desirable properties:

- The accuracy of the spatio-temporal image derivatives is not critical to the accuracy of the computation.
- The computed image flow is force-fitted to a model. The only conditions on the model are that it be space-limited and differentiable.
- With finite-extent basis functions ϕ_k, the system of equations gives a sparse, banded matrix structure.

2.2 Solution

In our experiments, we used the cosine window

$$\phi_0(x) = \frac{1}{2}\left[1 + \cos(\frac{\pi x}{w})\right] \quad x \in [-w, w] \tag{10}$$

as the prototype basis function. The entire basis was generated from shifts of the prototype along a rectangular grid with spacing w. This leads to an observation matrix $\hat{G}$ that is block tridiagonal:

$$\hat{G} = \begin{bmatrix} D_1 & U_1 & 0 & 0 & \\ L_2 & D_2 & U_2 & 0 & \cdots \\ 0 & L_3 & D_3 & U_3 & \\ & \vdots & & & \ddots \end{bmatrix} \tag{11}$$

where each of the submatrices D_i, U_i and L_i is in turn block tridiagonal, of the form

$$\begin{bmatrix} \times & \times & 0 & 0 & \\ \times & \times & \times & 0 & \cdots \\ 0 & \times & \times & \times & \\ & \vdots & & & \ddots \end{bmatrix} \tag{12}$$

and $\times$ denotes a 2-by-2 submatrix with data-dependent coefficients. In addition, $\hat{G}$ is block diagonal dominant, *almost* symmetric and *almost* positive semidefinite. In order to solve (6), we employ the method of *Preconditioned Biconjugate Gradients* (PBCG) [2; 8]. The structure of $\hat{G}$ allows for a good choice of useful preconditioners, one of which is the matrix $\tilde{G}$ formed by the even component of the diagonal 2-by-2 submatrices of $\hat{G}$. In effect, $\tilde{G}$ is the

component of $\hat{G}$ comprised purely of within-grid interactions.

Conjugate gradient methods are iterative algorithms for solving linear systems of the form $Ax = b$ by minimizing a quadratic functional such as $f(x) = \frac{1}{2}x'Ax - b'x$ over certain vector spaces called *Krylov spaces*. Each iteration adds, under ideal conditions, a dimension to the search space and generates an improved minimizer. When A is non-symmetric or indeterminate, a variant, the biconjugate gradient method, is used. Under the preconditioner $\tilde{G}$ the iterative equations take the form

$$\left.\begin{array}{rcl} \tau_k &=& \frac{\bar{r}_k'\tilde{G}^{-1}r_k}{\bar{d}_k'\hat{G}d_k} \\ x_{k+1} &=& x_k + \tau_k d_k \\ r_{k+1} &=& r_k + \tau_k \hat{G} d_k \\ \bar{r}_{k+1} &=& \bar{r}_k + \tau_k \hat{G}' d_k \\ \beta_k &=& \frac{\bar{r}_{k+1}'\tilde{G}^{-1}r_{k+1}}{\bar{r}_k'\tilde{G}^{-1}r_k} \\ d_{k+1} &=& -\tilde{G}^{-1}r_{k+1} + \beta_k d_k \\ \bar{d}_{k+1} &=& -\tilde{G}^{-1}\bar{r}_{k+1} + \beta_k \bar{d}_k \\ x_0 &=& \tilde{G}^{-1}\hat{A}\hat{b} \\ r_0 &=& \bar{r}_0 = \hat{G}x_0 - \hat{A}\hat{b} \\ d_0 &=& \bar{d}_0 = \tilde{G}^{-1}r_0 \end{array}\right\} \quad (13)$$

We have observed rapid convergence of the system, often to sufficient precision within 10 iterations even when the dimensionality of $\hat{G}$ is very large.

3 Stabilization

Stabilization is a differential process that compensates for the "unwanted" motion in the image sequence. In typical situations, the term "unwanted" denotes the motion in the sequence resulting from the kinematic motion of the camera with respect to an inertial frame of reference. In these situations, the "unwanted" component of the motion does not carry any information of relevance to the observer, and indeed strains its functioning. It can be shown that compensating for the full 3D motion of a 3D scene is tantamount to solving the structure-from-motion problem. It is possible to make simplifying assumptions on the structure of the scene to facilitate robust stabilization, which then involves (*i*) identifying an appropriate model to characterize the global motion, and (*ii*) robustly estimating the model parameters from the optical flow field.

3.1 Global Motion Model

Assume that a 3D scene is being imaged by a camera located at the origin with its optical axis along the Z axis. Let the camera translate with a linear velocity (t_x, t_y, t_z) and rotate with an angular velocity (w_x, w_y, w_z). The point $(X_i, Y_i, Z_i)'$ in 3-space is projected onto the image plane at $\mathbf{p}_i = (x_i, y_i)' = (f\frac{X_i}{Z_i}, f\frac{Y_i}{Z_i})'$, and moves with velocity

$$\dot{\mathbf{p}}_i = \begin{pmatrix} (-w_y x_i + w_x y_i)\frac{x_i}{f} + w_z y_i - \\ f(w_y - \frac{t_x}{Z_i}) + x_i \frac{t_z}{Z_i} \\ (-w_y x_i + w_x y_i)\frac{y_i}{f} - w_z x_i + \\ f(w_x - \frac{t_y}{Z_i}) + y_i \frac{t_z}{Z_i} \end{pmatrix} \quad (14)$$

Equation (14) is nonlinear in the unknowns $\{t_x, t_y, t_z, w_x, w_y, w_z, f, Z_i\}$ when there is non-zero translation and the $\dot{\mathbf{p}}_i$s are known at a sufficient number of points. The rotational and translational velocities cannot be computed (to within any reasonable degree of robustness) without computing the Z_is. The purpose of stabilization is not so much solving (14) as compensating for the effects of global camera motion. An assumption that is often made is that all unwanted motion is caused by the rotation of the camera, and the translations are small. When translation is significant, the formulation necessarily involves depth. Setting $t_x = t_y = t_z = 0$ yields a linear system in the unknowns (w_x, w_y, w_z) and a linearly constrained second-degree polynomial motion field in $\frac{x}{f}$ and $\frac{y}{f}$. For a normal lens, $\frac{x}{f} \leq 0.4$, the maximum being attained at the periphery of the image frame. It is therefore reasonable to assume the higher-order terms to be small given normal or telescopic optics. With such an assumption, the 3-parameter similarity transformation

$$\bar{\mathbf{p}}_i = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \mathbf{p}_i + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (15)$$

adequately models the global motion. An interesting case emerges under the assumption that all imaged points lie on a plane in 3D, *i.e.*

$$AX_i + BY_i + CZ_i = 1 \qquad \forall i \quad (16)$$

which is modeled by the 8-parameter projective transformation

$$\begin{pmatrix} \bar{x}_i \\ \bar{y}_i \end{pmatrix} = \frac{1}{c_1 x_i + c_2 y_i + 1} \begin{pmatrix} a_1 x_i + a_2 y_i + a_3 \\ b_1 x_i + b_2 y_i + b_3 \end{pmatrix} \quad (17)$$

With an increase in model order or the number of free parameters, the stabilized image sequence shows a smoother motion. However, the additional parameters often cause image distortions like warping of lines or an uneven expansion of the field of view. For our experiments, we have chosen an intermediate approach between the 3-parameter similarity transformation and the 8-parameter perspective model by using a 6-parameter affine model (18). Nevertheless, the procedure described here for robustly estimating model parameters holds for models of any degree of complexity

$$\bar{\mathbf{p}}_i = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \mathbf{p}_i + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (18)$$

3.2 Parameter Estimation

Since there may exist foreground areas that move quite differently from the background, the distribution of flow velocities, even when estimated without error, may be multimodal. Model parameter estimation involves locating the fundamental mode and its membership, a typical clustering problem with no mathematically concise solution. Also, although the optical flow is computed over the whole image, its reliability is local-gradient-dependent. Areas which have large gradients are typically, though not always, associated with more reliable flow estimates. In the first pass, grid points showing significant gradient content are picked out as "reliable" and the flow estimates at these points are combined in a least-squares framework to give a set of model parameters. The process is analogous to K-means clustering with the search for a cluster center being replaced by the search for model parameters. The model fit is compared to the local flow parameters and outliers based on sample statistics are discarded. This sieve is iterated a few times. When pruning the set of motion parameters used for computing the model parameters, the angular error measure employed in [3] is used. Assume that the true and computed flows at a point (x, y) in a particular frame are $(u_0, v_0)'$ and $(u, v)'$ respectively. Define vectors $\mathbf{v}_0 = (u_0, v_0, 1)'$ and $\mathbf{v} = (u, v, 1)'$. The error angle ϵ at (x, y) is given by

$$\epsilon = \arccos\left(\frac{\mathbf{v}_0 \cdot \mathbf{v}}{\|\mathbf{v}_0\|\|\mathbf{v}\|}\right) \tag{19}$$

ϵ is insensitive to the magnitude of the motion vector and offers a normalized measure against which a range of velocities can be compared meaningfully. The final model estimates are obtained after a few iterations of pruning this set while ensuring that a sufficient number of grid points remain.

3.3 Gross Motion Compensation

The first phase of stabilization consists of compensating for large, temporally correlated motion between frames. The image sequence is suitably downsampled and temporal gradients are computed over a long support (typically 7 frames). The robust optical flow model is estimated, and from it the global motion model parameters are computed. Since there is no feedback to complete the loop and check for overall compliance, there is typically a small residual motion in the stabilized frames. This motion is uncorrelated between frames.

3.4 Jitter Removal

Most electronic image stabilizers, whether feature-based or flow-based, leave behind a small residual motion in the stabilized frames. Since the motion is small and temporally uncorrelated, we can use the robust optical flow algorithm with temporal derivatives computed over two frames. The image sequence at this stage is largely stabilized. Assuming that we have an internal representation of the "pristine" stabilized frame, we can compute spatial derivatives from it. This pristine frame is none other than the mosaic we build from the stabilized sequence. To bootstrap the system, the first frame input to this block forms the first mosaic image for this block. As successive frames are fed in, the jitter is computed, the input frames are rewarped and the mosaic is updated taking the warping parameters into account.

4 Mosaicking and Super-resolution

The light pattern falling on the imaging surface of the camera is a continuous function of space and time, that is discretized and sampled to form the image sequence. For the sake of brevity, we will assume the exposure to be instantaneous in the following discussion. Let $l(x, y)$ denote the incident light intensity at the current instant, which is sampled by $s(\cdot, \cdot)$ to give the sampled intensity at pixel (i, j) as the continuous-domain convolution

$$I(i, j) = (i, j) \otimes s(i, j) \tag{20}$$

In a dynamic 3D scene, factors influencing $l(x, y)$ include occlusion, motion of foreground objects and change in pose, as well as the background luminance pattern and sensor noise. Let $f(x, y)$ denote the luminance pattern of the background at $t = 0$. We can write $l(x, y)$ as a sum of a shifted $f(x, y)$ and terms for the effects of foreground objects, occlusion and sensor noise, grouped into $\eta(x, y)$. Let the background shift be (x_s, y_s). We assume, for simplicity, that the shift is constant throughout the image, although it is possible to handle the case of space-varying shift with some added complexity. The imaged intensity pattern is

$$l(x, y) = f(x + x_s, y + y_s) + \eta(x, y) \tag{21}$$

In the sampled domain the intensity pattern $I(x, y)$ is given by

$$\begin{aligned} I(i, j) &= [f_0(i + x_s, j + y_y) + \eta(i, j)] \otimes s(i, j) \\ &= [f_0(i, j) + \eta_s(i, j)] \otimes s(i + x_s, j + y_s) \end{aligned} \tag{22}$$

The process of rewarping an image by a non-integral shift essentially involves the following steps: (i) the sampled input image $I(i, j)$ is converted to a continuous spatial function $I(x, y)$, which is (ii) shifted by $(-x_w, -y_w)$ and (iii) resampled to give $I_s(x, y)$:

$$\begin{array}{rcl} I(x, y) & = & \sum_{\hat{i}} \sum_{\hat{j}} I(\hat{i}, \hat{j}) h(x - \hat{i}, y - \hat{j}) \\ I_s(x, y) & = & I(x, y) \otimes \delta(x + x_w, y + y_w) \\ I_s(i, j) & = & \sum_{\hat{i}} \sum_{\hat{j}} I(\hat{i}, \hat{j}) \\ & & h(i + x_w - \hat{i}, j + y_w - \hat{j}) \end{array} \tag{23}$$

Assume that the warping parameters (x_w, y_w) are known to within a small error, *i.e.*

$$x_w = x_s + \delta_x \qquad y_w = y_s + \delta_y \tag{24}$$

Equations (22), (23) and (24) give

$$\begin{aligned} I_s(i,j) &= \textstyle\sum_{\hat{i}} \sum_{\hat{j}} [f_0(i,j) + \eta_s(i,j)] \otimes \\ & s(i + x_s, j + y_s) h(i - x_w - \hat{i}, j - y_w - \hat{j}) \\ &= [f_0(i,j) + \eta_s(i,j)] \otimes w(i + \delta_x, j + \delta_y) \end{aligned} \tag{25}$$

where $w(\cdot,\cdot) = s(\cdot,\cdot) * h(\cdot,\cdot)$ is the composite of the imaging and warping processes, calculated by a discrete convolution with a possibly non-integral shift.

Equation (25) can be written as

$$\begin{aligned} I_s(i,j) &= f_0(i,j) \otimes w(i,j) + \\ (\delta_x \delta_y)' \nabla (f_0(i,j) \otimes w(i,j)) &+ o(\delta_x, \delta_y) + \\ \eta_s(i,j) &\otimes w(i + \delta_x, j + \delta_y) \end{aligned} \tag{26}$$

The first term is independent of time. Errors in estimating the true shift lead to differential terms. In the above equation, the statistics of the other terms are largely indeterminate. In practice, the noise components are non-Gaussian and heavy-tailed. In order to perform temporal averaging to estimate the steady term, it is necessary to apply a robust estimator like an order-statistics filter. We have used a sliding-window median filter for estimating the background intensity from the temporal stack of stabilized images.

In building a higher-resolution image from the data gathered by the sensor, the rewarping mechanism is upsampled by an appropriate factor. The same reasoning applies here as for mosaicking, to justify using a robust temporal filter in order to estimate the constant component of the signal corresponding to the perfectly stabilized background. The final task is to deconvolve the resulting image using the known composite blur $w(\cdot,\cdot)$.

5 Results

Fig.2 shows the performance of the algorithm on data gathered by a ground vehicle ((a) and (c)) and by an airborne camera ((b), (d)-(f)). The first frames of the two sequences are shown in Figs. 2 (a) and (d) respectively. In the latter case, the image sequence itself was regenerated from an MPEG-compressed file that was available to us. The backward-stabilized image of the former sequence is shown in Fig.2(c). Here, the inner rectangle corresponds to the latest viewing area and the older frames are warped back with respect to the current frame. The 3D nature of the world, as it evolves in time, does not permit a topologically correct embedding in the 2D plane of the image. Nevertheless, the mosaic does convey visually meaningful information, *e.g.* the presence of foreground trees on either side, and the overall rightward path of the vehicle (which is seen from the uneven distortion in the field of view and also by the right arc formed by the locus of the "hood ornament"). In the sequence gathered from the aerial platform, the targets on the ground are moving vehicles which are on the order of a single pixel in size, and move with fractional pixel velocity. The mosaic reconstructed from the first minute (280 frames) is shown in Fig.2(b) and the super-resolved image reconstructed from the first 20 frames in Fig.2(c). Fig.2(d) shows the difference image between the 20th frame and the super-resolved mosaic. The moving vehicles show up as bright spots.

References

[1] P. Anandan *et al.*, "Real-time Scene Stabilization and Mosaic Construction", *DARPA Image Understanding Workshop*, 1994.

[2] O. Axelsson, *Iterated Solution Methods*, Cambridge University Press, 1994.

[3] J. L. Barron, D. J. Fleet and S. S. Beauchemin, "Performance of Optical Flow Techniques", *Int. Jour. of Comp. Vision*, vol. 12, pp. 43–77, 1994.

[4] P. J. Burt and P. Anandan, "Image Stabilization by Registration to a Reference Mosaic", *DARPA Image Understanding Workshop*, 1994.

[5] O. J. Kwon, R. Chellappa and C. H. Morimoto, "Motion-Compensated Subband Coding of Video Acquired from a Moving Platform", *IEEE ICASSP*, pp. 270–275, 1995.

[6] M. Irani and S. Peleg, "Improving Resolution by Image Registration", *Graphical Models Image Proc.*, vol. 53, pp. 231–239, 1991.

[7] C. H. Morimoto and R. Chellappa, "Fast Electronic Digital Image Stabilization", *ICPR*, vol. 3, pp. 284–288, 1996.

[8] W. H. Press *et. al.*, *Numerical Recipes in C* (2nd ed.), Cambridge University Press, 1992.

[9] S. Srinivasan and R. Chellappa, "Robust Modeling and Estimation of Optical Flow with Overlapped Basis Functions", CAR-TR-845, University of Maryland, 1996.

[10] Y. S. Yao, "Electronic Stabilization and Feature Tracking in Long Image Sequences", CAR-TR-790, University of Maryland, 1996.

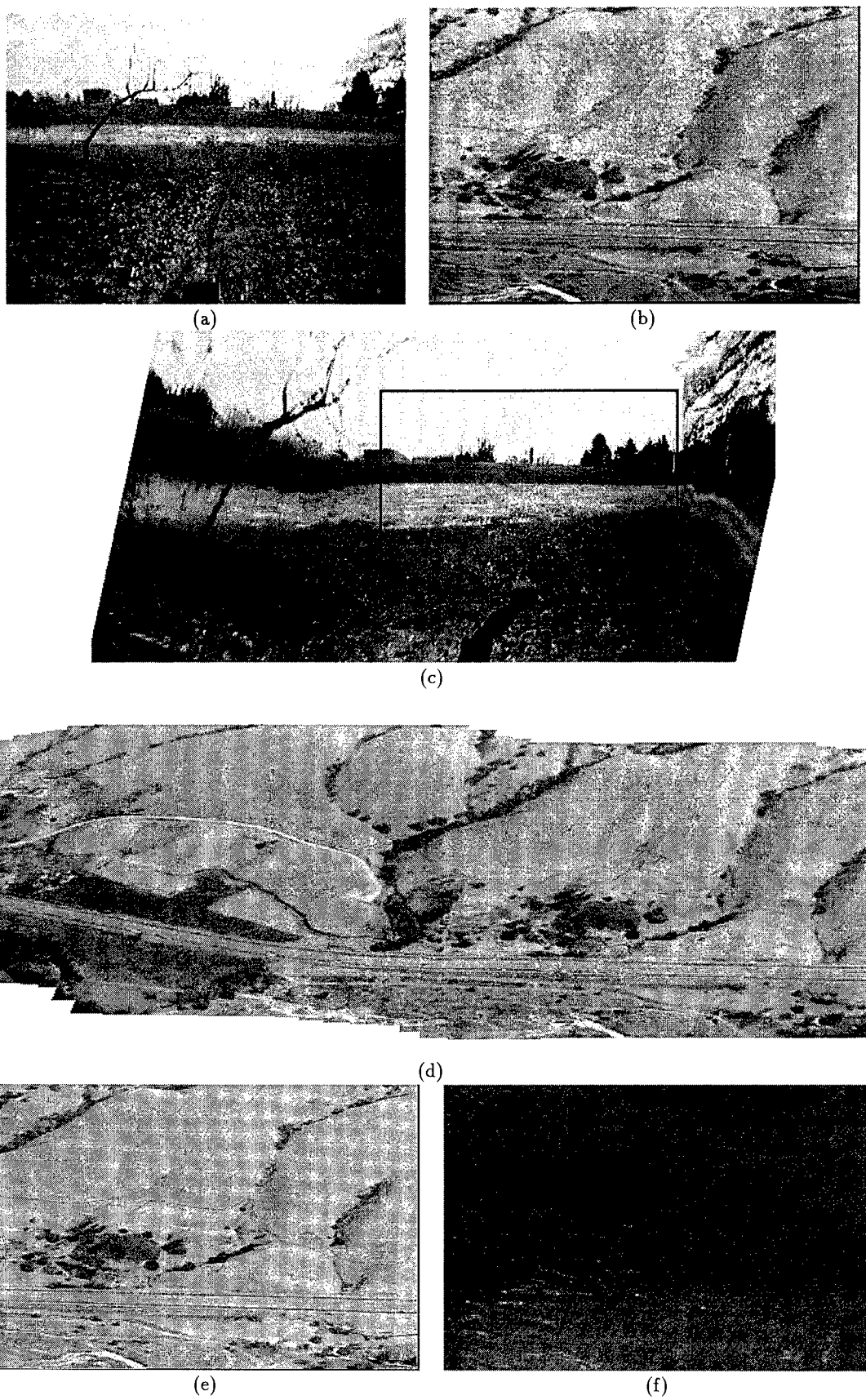


Figure 2: Results: (a) First frame of MM sequence acquired from ground vehicle; (b) first frame of S69 sequence of aerial imagery, decompressed from an MPEG stream; (c) stabilized mosaic of 100 frames of MM sequence showing an expanded field of view and distortions due to 3D effects; (d) mosaic of first 280 frames of S69; (e) super-resolved image from first 20 frames; (f) difference image showing target locations as bright spots.

Mosaicing with Generalized Strips *

Benny Rousso Shmuel Peleg Ilan Finci
Inst. of Computer Science
The Hebrew University of Jerusalem
91904 Jerusalem, ISRAEL

Abstract

Video mosaicing is commonly used to increase the effective visual field of view. Existing mosaicing methods are based on image alignment, and are effective only in very limited cases.

To overcome most restrictions, mosaicing is presented in this paper as a process of collecting strips. Strips which are perpendicular to the optical flow are cut out of the images, and are warped so that within each strip the optical flow will be parallel. These strips are then pasted into the mosaic. This approach enables to define mosaicing even for cases of forward motion and for zoom. View interpolation, generating dense intermediate views, is used to overcome parallax effects.

1 Introduction

An introduction and a survey of mosaicing methods can be found in [6]. We will only give a brief introduction focusing on the aspects relevant to this work.

Early mosaicing methods were used for aerial and satellite images. In both cases the objects in the scene are distant from the camera, and camera motion could be modeled as a translation parallel to the image plane with no parallax effects. Other methods use a camera which is rotating around the y axis passing through it's optical center without any translation. The resulting mosaic corresponds to a projection onto a cylinder [5].

Most methods select one frame to be a reference frame, towards which all other frames are warped [8, 3]. This approach uses 2D analysis to find motion based on an affine model or a general planar surface model, and allows somewhat more general camera motion. However, this approach can not handle parallax, and is restricted to small rotations (around the x or the y axis) with regard to the reference frame. Large rotations cause distortions when trying to perform the reprojection onto the reference frame. In addition, existing methods are not well defined for forward motion or for zoom.

To overcome most restrictions, mosaicing is defined here as a process of collecting strips from image sequences satisfying the following conditions:

- Strips should be perpendicular to the optical flow.
- The collected strips should be warped and pasted into the panoramic image such that when warping their original optical flow it becomes parallel to the direction in which the panoramic image is constructed.

Using these properties, we define mosaicing methods for the case of 2D affine motion. This covers most simple cases, and also zoom and forward motion. Generated mosaics have minimal distortions compared to the original images, as no global scaling is performed.

The strip collection process allows the introduction of a mechanism to overcome the effects of parallax by generating dense intermediate views. In many cases mosaics generated in this manner can be considered at linear pushbroom cameras [2].

2 Mosaicing Using Strips

Construction of panoramic mosaics includes the collection of sections from each image and pasting these sections next to each other to become the mosaic. In the simple case of a camera which is moving horizontally, vertical sections are usually taken from each image and pasted side by side (see Fig. 1.a). In this case the process can also be viewed as scanning the scene with a vertical line. This vertical line scans the entire sequence, extracts vertical strips along the sequence, and pastes them one next to the other to create the panoramic mosaic. In this case the vertical line is perpendicular to the horizontal optical flow, and after placing the strips in the panoramic image, the optical

*This research was partially funded by DARPA through the U.S. Office of Naval Research under grant N00014-93-1-1202 and by the European ACTS project AC074 "Vanguard". Contact E-Mail: peleg@cs.huji.ac.il

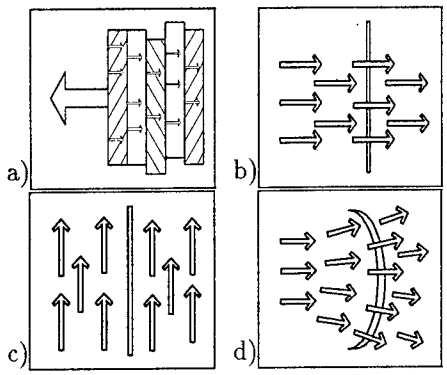


Figure 1: The relation between the mosaicing process and the direction of the Optical Flow. (a) The simple case of camera which is moving to the left. The Optical Flow points to the right, and vertical strips are collected. After pasting, The Optical Flow is parallel with the direction in which the panoramic image is built. (b) New information is passing through a given line when the Optical Flow is perpendicular to the line. (c) No new information is passing through a given line when it is not perpendicular to the Optical Flow. (d) In the general case the line is set to be perpendicular to the Optical Flow.

flow is pointing exactly to the direction from which the panoramic image is constructed (see Fig. 1.b).

Using such a vertical scanning line with vertical camera motion, when the optical flow is parallel to this scanning line, (see Fig. 1.c), will not create any mosaic, as no new information will pass through the selected line.

In general, optimal results would be achieved by selecting a scanning line which is perpendicular to the optical flow (see Fig. 1.d). The information from all images in the sequences will pass through the scanning line, allowing to collect strips for pasting in the mosaic.

The requirement that the scanning line be perpendicular to the optical flow can be described for a pair of subsequent images I_{n-1} and I_n. If a point $p_n = (x_n, y_n)$ in I_n is on the scanning line, and corresponds to point $p_{n-1} = (x_{n-1}, y_{n-1}) = (x_n - u, y_n - v)$ in image I_{n-1}, then new information arrive to the point p_n from the direction $(-u, -v)$, and for optimal results the direction of scanning line at point p_n should be perpendicular to $(-u, -v)$.

2.1 Affine Motion

In many cases the motion between two images is approximated by an affine transformation. Many methods exist to recover the parameters of an affine transformation [4].

The affine transformation can be expressed as:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} x_n - x_{n-1} \\ y_n - y_{n-1} \end{pmatrix} = \begin{pmatrix} a + bx_n + cy_n \\ d + ex_n + fy_n \end{pmatrix} \quad (1)$$

where (x_{n-1}, y_{n-1}) and (x_n, y_n) are corresponding points in images I_{n-1} and I_n, and the parameters of the affine transformation $\mathcal{A}$ are (a, b, c, d, e, f). (u, v) is the optical flow vector as a function of the position (x_n, y_n). The transformation $\mathcal{A}$ (and the optical flow) vary continuously along the sequence.

We are looking for a line $\mathcal{F}(x, y) = 0$ such that it will be perpendicular to the optical flow. The normal to the line $\mathcal{F} = 0$ is in the direction $(\frac{\partial \mathcal{F}}{\partial x}, \frac{\partial \mathcal{F}}{\partial y})$, thus it should be in the same direction as (u, v). This constraint can be expressed by:

$$\begin{pmatrix} \frac{\partial \mathcal{F}}{\partial x} \\ \\ \frac{\partial \mathcal{F}}{\partial y} \end{pmatrix} = k \begin{pmatrix} u \\ \\ v \end{pmatrix} = k \begin{pmatrix} a + bx + cy \\ \\ d + ex + fy \end{pmatrix} \quad (2)$$

for some value of k. By integrating, we get the equation of the scanning line:

$$0 = \mathcal{F}(x, y) = ax + dy + \frac{b}{2}x^2 + \frac{c+e}{2}xy + \frac{f}{2}y^2 + M \quad (3)$$

This is a family of lines that are all perpendicular to the optical flow. M is used to select a specific line. We suggest that M will be set to the value for which the line contains maximum number of pixels within the image. If many options exit, then we suggest using a line as close as possible to the center of the image to minimize lens distortions.

Note that this line equation exists only when $e = c$. In most cases, the difference between the values of c and e is due to the rotation around the optical axis ω_z, such that it contributes $-\omega_z$ to c, and $+\omega_z$ to e. As a result, the term $\frac{c+e}{2}$ should be approximately equal to the common component of c and e, which excludes the rotations around the optical axis. As rotation around the optical axis does not expose any new information regarding the scene, effects of such a rotation should be eliminated by preliminary warping of the image with the rotation $\omega_z \approx \frac{e-c}{2}$, which is known once the affine transformation is recovered.

We will use the following notation to describe the scanning line along the sequence: The line $\mathcal{F}_n(x_n, y_n) = 0$ is the line in Image I_n, in it's coordinate system (x_n, y_n), which corresponds to the affine transformation $\mathcal{A}_n = (a_n, b_n, c_n, d_n, e_n, f_n)$. This affine transformation $\mathcal{A}_n$ relates points p_n in Image I_n to corresponding points p_{n-1} in Image I_{n-1} (see Fig. 3).

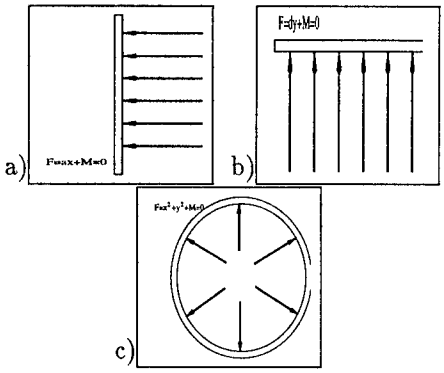


Figure 2: Examples for scanning line.
(a) A vertical scanning line is selected for horizontal motion. (b) A horizontal scanning line is selected for vertical motion. (c) A circular scanning line is selected for zoom and for forward motion.

2.2 Special Cases

Eq. 3 can be easily understood for some simple cases.

- In the case of sideway motion (either small sideway rotation or sideway translation), the affine transformation $\mathcal{A}$ takes the form $\mathcal{A} = (a, 0, 0, 0, 0, 0)$, thus the selected line becomes $0 = \mathcal{F}(x, y) = ax + M$, which is a vertical line (see Fig. 2.a).
- In the case of upwards motion (either small rotation or translation), the affine transformation takes the form $\mathcal{A} = (0, 0, 0, d, 0, 0)$, thus the selected line becomes $0 = \mathcal{F}(x, y) = dy + M$, which is a horizontal line (see Fig. 2.b).
- In the case of zooming or forward motion (towards a planar surface which is parallel to the image plane), the affine transformation takes the form $\mathcal{A} = (0, s, 0, 0, 0, s)$, where s is the scaling factor. As a result, the selected line will become $0 = \mathcal{F}(x, y) = \frac{s}{2}(x^2 + y^2) + M$, which is a circle around the center of the image (see Fig. 2.c).

In the more general translation case, the result will be a circle around the Focus Of Expansion (FOE), assuming that the scene is planar and parallel to the image plane. More complex cases exist, in which the result will be generalized elliptic curve.

3 Cutting and Pasting of Strips

The mosaic is constructed by pasting together strips taken from the original images. The shape of the strip, and its width, depend on the image motion. This section describes how to select and to paste these strips.

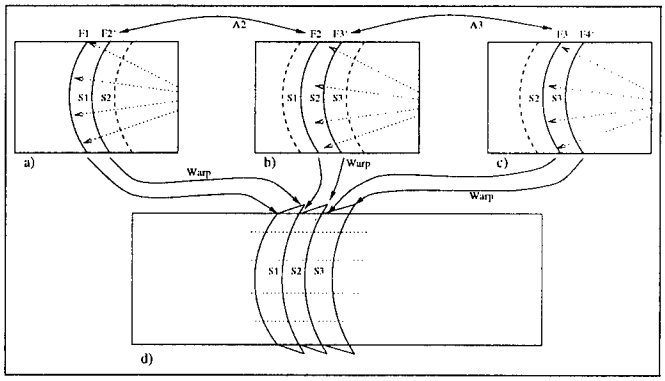


Figure 3: Cutting and pasting strips.
(a)-(c) Strips are perpendicular to the optical flow. (d) Strips are warped and pasted so that their back is fixed and their front is warped to match the back of the next strip.

3.1 Cutting Strips

In order to determine the strip to be taken from Image I_n, the preceding frame, I_{n-1}, and the succeeding frame, I_{n+1}, should be considered.

Let $\mathcal{A}_n$ be the affine transformation relating points $p_n = (x_n, y_n)$ in Image I_n to the corresponding points $p_{n-1} = (x_{n-1}, y_{n-1})$ in Image I_{n-1}, and let $\mathcal{A}_{n+1}$ be the affine transformation relating points $p_{n+1} = (x_{n+1}, y_{n+1})$ in Image I_{n+1} to the corresponding points $p_n = (x_n, y_n)$ in Image I_n.

Given the affine transformations $\mathcal{A}_n$ and $\mathcal{A}_{n+1}$, the lines $\mathcal{F}_n(x_n, y_n) = 0$ and $\mathcal{F}_{n+1}(x_{n+1}, y_{n+1}) = 0$ are selected respectively (see Fig. 3.a-c). The line $\mathcal{F}_n(x_n, y_n) = 0$ in I_n corresponds to the line $\mathcal{F}'_n(x_{n-1}, y_{n-1}) = 0$ in I_{n-1} using the affine transformation $\mathcal{A}_n$. In the same way, the line $\mathcal{F}_{n+1}(x_{n+1}, y_{n+1}) = 0$ in I_{n+1} corresponds to the line $\mathcal{F}'_{n+1}(x_n, y_n) = 0$ in I_n using the affine transformation $\mathcal{A}_{n+1}$.

The strip that is taken from the image I_n is the range between the lines $\mathcal{F}_n(x_n, y_n) = 0$ and $\mathcal{F}'_{n+1}(x_n, y_n) = 0$ in I_n (see Fig. 3.a-c).

Using this selection, the first boundary of the strip will be described by the selected line $\mathcal{F}_n$, thus will be exactly orthogonal to the optical flow with regard to the previous image. The second boundary of the strip is described by the line $\mathcal{F}'_{n+1}$ which is the projection of the line $\mathcal{F}_{n+1}$ onto the current image I_n, having the same properties in the next image.

This selection of the boundaries of the strip ensures that no information is missed nor duplicated along the strip collection, as the orthogonality to the optical flow is kept.

3.2 Pasting Strips

Consider the common approach to mosaicing where one of the frames is used as a reference frame, and all other frames are aligned to the reference frame before pasting. In term of strips, the first strip is put in the panoramic image as is. The second strip is warped in order to match the boundaries of the first strip. The third strip is now warped to match the boundaries of the *already warped* second strip, etc. As as result, the mosaic image is continuous. However, major distortions may be caused by the accumulated warps and distortions. Large rotations can not be handled, and cases such as forward motion or zoom usually cause unreasonable expansion (or shrinking) of the image.

To create continuous mosaic images while avoiding accumulated distortions, the warping of the strips should not be done towards the mosaic, but towards another original frame. In our scheme, the back of each strip is never changed. This is the side of the strip which corresponds to the boundary between Image I_{n-1} and Image I_n and defined by $\mathcal{F}_n$. The front of the strip is warped to match the back side of the next strip. This is the boundary between Image I_n and Image I_{n+1} which is defined by $\mathcal{F}'_{n+1}$.

In the example described in Fig. 3.d, we warp the first strip such that its left side does not change, while its right side is warped to match the left side of the original second strip. In the second strip, the left side does not change, while the right side is warped to match the left side of the third strip, etc.

As a result, the constructed image is continuous. Also, were we to warp the original optical flow as we did with the strips, the resulting flow is continuous as well, and is parallel to the direction in which the panoramic mosaic is constructed. Moreover, no accumulative distortions are encountered, as each strip is warped to match just another original strips, avoiding accumulative warps.

4 View Interpolation for Parallax

Taking strips from different images when the width of the strips is more than one pixel would work fine only without parallax. When parallax is involved, no single transformation can be found to represent the optical flow in the entire scene. As a result, a transformation that will align a close object will duplicate far objects, and on the other hand, a transformation that will align a far object will truncate closer objects. Also, rapid changes between aligning close and far objects might result in useless results.

In order to overcome the parallax problems in general scenes, instead of taking a strip with a width of L pixels, we can synthetically generate intermediate images, and use narrower strips. For example, we can take a collection of L strips, each with a width of one pixel, from interpolated camera views in between the original camera positions. In order to synthesize new views we can use various methods, such as optical flow interpolation [1, 9], trilinear tensor methods [7], and others. In most cases approximate methods will give good results. The creation of the intermediate views can involve only view interpolation, as in most of the applications view extrapolation is not needed.

The use of intermediate views for strips collection gives the effect of orthographic projection, which avoids parallax discontinuities. This strategy can be combined with the methods that were described in the previous sections as a preliminary stage, such that a complete solution is given for general motion in general scenes.

5 Experimental Results

In this section we show two cases which can not be done with other mosaicing methods. These results are still preliminary, but indicate the potential of this approach. Simple cases can be seen in [6].

5.1 Zoom

During zoom, the resolution of the image increases while the field of view becomes smaller, causing the loss of the outside periphery from the next frame. Our process collects these circular peripheral strips, that disappear from one frame to the next, to construct the mosaic.

Assume the camera is located at the side of a long wall, with its optical axis parallel to the wall. In this case the closest parts of the wall are seen in high details at the edge of the image, while the distant parts of the wall are seen smaller closer to the center of the image. When zooming in, the further parts are magnified and get closer to the edge of the image, and the mosaic will therefore become a reconstruction of the wall at the highest possible resolution. Under some conditions the wall can even be reconstructed as viewed from the front, in uniform resolution all over. This result is shown in Fig. 4, where circular strips were collected and pasted in the panoramic image.

5.2 Sideway Motion with Parallax

In Figure 5 the camera is moving sideways, generating substantial parallax. Vertical strips were collected according to the affine transformation that was recovered along the sequence, and the strips were pasted in the panoramic image. Without view interpolation, duplications and truncations are seen clearly, while with view interpolation these effects are reduced. The view interpolation was performed by optical flow interpolation.

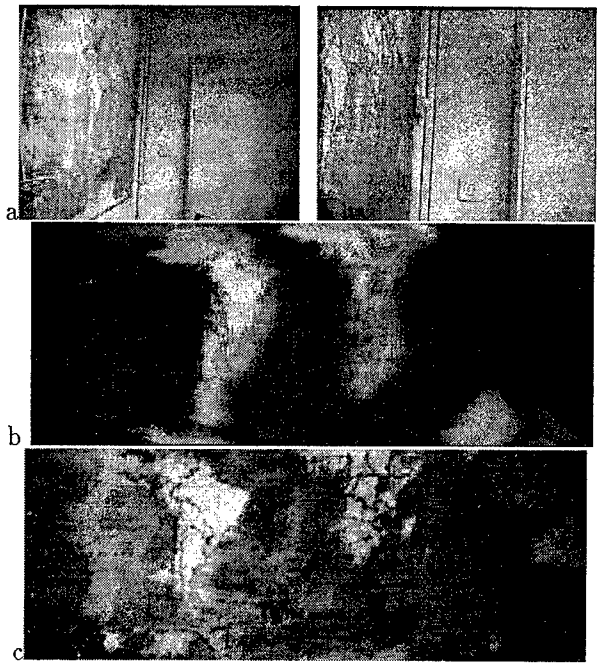


Figure 4: Panoramic mosaic for Zoom.
(a) Two original images. A map is seen on a wall parallel to the optical axis. (b) Reconstructed panoramic mosaic, which is similar to a real front view of the map (c).

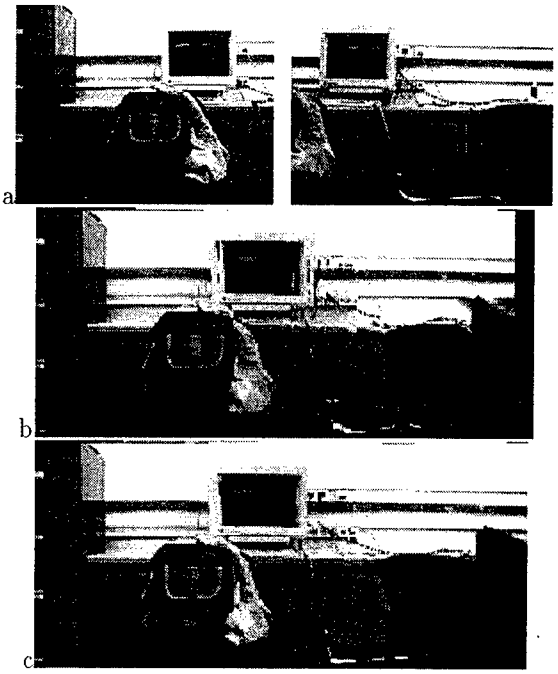


Figure 5: Handling parallax: sideways motion
(a) Two original images. (b) Mosaicing without view interpolation. Distant objects are duplicated, and close objects are truncated. (c) Using view interpolation reduces the distortions.

Bibliography

[1] S.E. Chen and L. Williams. View interpolation for image synthesis. In *SIGGRAPH*, pages 279–288, Anahiem, California, August 1993. ACM.

[2] R. Hartley and R. Gupta. Linear pushbroom cameras. In J.O. Eklundh, editor, *Third European Conference on Computer Vision*, pages 555–566, Stockholm, Sweden, May 1994. Springer.

[3] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *Fifth International Conference on Computer Vision*, pages 605–611, Cambridge, MA, June 1995. IEEE-CS.

[4] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In G. Sandini, editor, *Second European Conference on Computer Vision*, pages 282–287, Santa Margherita, Italy, May 1992. Springer.

[5] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *SIGGRAPH*, Los Angeles, California, August 1995. ACM.

[6] S. Peleg and J. Herman. Panoramic mosaics with VideoBrush. In *IUW-97*, New Orleans, Louisiana, May 1997. Morgan Kaufmann.

[7] B. Rousso, S. Avidan, A. Shashua, and S. Peleg. Robust recovery of camera rotation from three frames. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 796–802, San Fransisco, California, June 1996.

[8] H.S. Sawhney, S. Ayer, and M. Gorkani. Model-based 2D & 3D dominant motion estimation for mosaicing and video representation. In *Fifth International Conference on Computer Vision*, pages 583–590, Cambridge, MA, June 1995. IEEE-CS.

[9] S. Seitz and C. Dyer. Physically valid view synthesis by image interpolation. In *Proc. IEEE Workshop on Representation of Visual Scenes*, Cambridge, MA, June 1995. IEEE-CS.

Panoramic Mosaics with VideoBrush™ *

Shmuel Peleg
Inst. of Computer Science
The Hebrew University
91904 Jerusalem, ISRAEL

Joshua Herman
David Sarnoff Research Center
CN 5300
Princeton, NJ 08540, USA

Abstract

As the field of view of a picture is much smaller than our own visual field of view, it is common to paste together several pictures to create a panoramic mosaic having a larger field of view. While scissors and glue are the tools used in film photography, more sophisticated methods were enabled with digital video.

Panoramic mosaics can be created by special devices which rotate around the camera's optical center (Quicktime VR, Surround Video), or by aligning, and pasting, frames in a video sequence to a single reference frame. Existing mosaicing methods have strong limitations on imaging conditions, and distortions are common.

Manifold projection enables the creation of panoramic mosaics from video sequences under very general conditions. The panoramic mosaic is a projection of the scene into a virtual manifold whose structure depends on the camera's motion. This manifold is more general than the customary projections onto a single image plane or onto a cylinder. VideoBrush, which is a real-time, software only, implementation on a PC, proves the superior quality and speed of this approach.

1 Introduction

The need to combine pictures into panoramic mosaics existed since the beginning of photography, as the camera's field of view is always smaller than the human field of view.

Three major issues are important in image mosaicing:

- Image alignment, which determines the transformation that aligns the images to be combined into a mosaic.
- Image cut and paste is necessary since most regions in the panoramic mosaic are overlapping, and are covered by more than one picture.
- Image blending is necessary to overcome the intensity difference between images, differences that are present even when images are perfectly aligned.

The simplest mosaics are created from a set of images whose mutual displacements are pure image-plane translations. This is approximately the case with some satellite images. Other simple mosaics are created by rotating the camera around its optical center using a special device, and creating a panoramic image which represents the projection of the scene onto a cylinder [7, 15, 14, 13]. Since it is not simple to ensure a pure rotation around the optical center, such mosaics are used only in limited cases.

In more general camera motions, that may include both camera translations and camera rotations, more general transformation for image alignment are used [5, 8, 12, 16, 9]. In all cases images are aligned pairwise, using a parametric transformation like an affine transformation or planar-projective transformation. A reference frame is selected, and all images are aligned with this reference frame and combined to create the panoramic mosaic.

Aligning all frames to a single reference frame is reasonable when the camera is far away and its motion is mainly a translation and a rotation around the optical axis. Significant distortions are created when camera motions includes other rotations.

Manifold Projection overcomes many of the difficulties in photo-mosaicing:

- The projection is defined for almost any arbitrary camera motion and any scene structure. This is enabled by narrowing the goal of image alignment from perfect alignment of all overlapping image regions to alignment only along the seam between the images.
- There are no distortions caused by the alignment to a reference frame. Object size in the panoramic

*This research was mostly done at David Sarnoff Research Center, Princeton, NJ, USA. VideoBrush is a trademark of Sarnoff. Contact E-Mail: peleg@cs.huji.ac.il

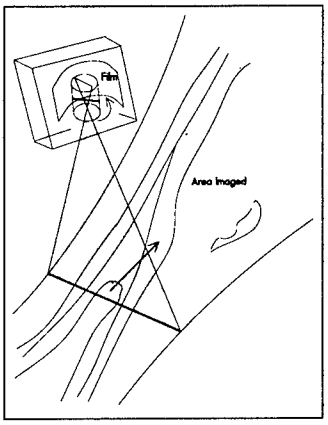


Figure 1: Aerial photography with a 1D scan system.

mosaic is the same as in the original images, and therefore the resolution in the mosaic is the same as the image resolution.

- Computation is simplified as the only image warping used are image-plane translations and rotations.

VideoBrush is an initial implementation of Manifold Projection which performs real-time mosaicing from a video sequence on a PC without any hardware acceleration.

2 Manifold Projection

Manifold Projection simulates the sweeping of the scene with a plane using a one-dimensional sensor array (Figure 1). Such a 1-D sensor can scan the scene by arbitrary combinations of rotations and translations, and in all cases the scanning will result in a sensible panoramic image if we could figure out how to align the incoming 1D image strips. Some satellite images are created by scanning the earth with a 1-D sensor array using a rotating mirror. Since in this case the alignment of the sensors can be done using the location of the satellite and the position of the mirror, panoramic 2D images are easily obtained. Figure 1 is an example of such a 1D scan system.

In more general cases the motion of the sweeping plane may not be known. It seems impossible to align the 1-D image strips coming from an arbitrary plane sweep, but the problem becomes easier when the input is a video sequence. A 2D frame in a video sequence can be regarded as having a 1-D strip somewhere in the center of the image ("center strip"), embedded in the 2D image to facilitate alignment. The motion of the sweeping plane can then be computed from the entire image, and applied on the center-strip for alignment and mosaicing.

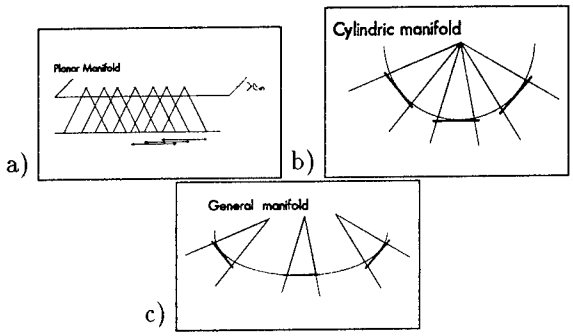


Figure 2: Different cases in Manifold Projection, where the projection is onto a smooth manifold passing through the centers of the image planes used for mosaicing. The camera is located at the tip of the "field-of-view" cone, and the image plane is marked by a bold segment.
a. Pure camera translation: parallel projection onto a plane.
b. Pure camera rotation: projection onto a cylindrical manifold.
c. Combined translation and rotation: the manifold is not simple any more.

The image transformations of the 1D strips generated by the sweeping plane are only rigid transformations: image plane translations and rotations. Therefore, rigid transformations should also be the transformation used in Manifold Projection. It should be noted that general camera motions induce, in general, non-rigid image-plane transformations. However, to simulate the plane sweep only rigid transformations should be used for the center-strip.

The panoramic mosaics generated by combining the aligned 1D center-strips form a new scene-to-image projection, called the Manifold Projection. This is a projection of the scene into a general manifold which is a smooth manifold passing through centers of all image planes constructing the mosaic. In the case of pure camera translations (Figure 2.a), Manifold projections turns out to be a parallel projection onto a plane. In the case of pure camera rotations (Figure 2.b), it is a projection onto a cylinder. But when both camera translations and rotations are involved, as in Figure 2.c, the manifold is not a simple manifold any more. The ability to handle such arbitrary combinations of camera rotations and translations is the major distinction between Manifold Projection and all previous mosaicing approaches.

The type of camera motion has a very significant impact on the type of projection and on the appearance of the panoramic mosaic. In camera panning,

where the camera motion is a pure rotation around the Y-axis, the resulting projection is onto a cylinder. This generates a mosaic which is, locally, very similar to every input image.

In a pure camera translation, where the camera moves parallel to the image plane, manifold projection is a semi-parallel projection onto a plane. Semi-parallel means that each center-strip is parallel to the other center-strips, but within the center-strips the projection is still perspective. Parallel projection is very different from a perspective projection in the sense that far-away objects do not appear smaller than close-by objects.

3 Image Alignment

Simulation of scene sweeping by a plane from a given video sequence can be done once the full 3D motion of the camera ("ego-motion") is known [11]. However, the implementation of the manifold projection described in this paper uses only 2D alignment, rather than using full ego-motion analysis. Nevertheless, results are impressive in most cases. It has most of the desired features of the theoretical manifold projection, e.g. that each object in the mosaic appears in the same shape and size as it appears in the video frames, avoiding any scaling, and therefore avoiding the possible associated distortions and loss of resolution. The 2D alignment used therefore compensates only for image translations and rotations. Another assumption in this implementation is that scale changes are minimal: there is no change of focal length, and the effects of forward motion are significantly smaller than the effects of other motions.

To assure that the motion computation will always result in the image motion of a single object, methods similar to [10, 6] were used.

4 Cut and Paste

Combination of the sequence of aligned image frames into a single panoramic mosaic can be done in several ways. In those cases where image alignment is close to perfect, pixel values in the panoramic mosaic can be computed by averaging the corresponding values in all overlapping pixels of the aligned original frames.

When the alignment between images is not perfect, averaging may result in blurring and in deterioration of image quality. In this case it is preferred to select only one of the input images to represent a region in the mosaic. Such a selection should be done to minimize effects of misalignment. The most logical selection is to select from each image that part closest to its center. There are two reasons for that selection:

- Alignment is usually better at the center than at the edges of the pictures.
- Image distortion is minimal at the center of the images.

This selection corresponds to the Voronoi tessellation [3]. Using the Voronoi tessellation for image cut-and-paste also served to minimize visible misalignment due to lens distortions. Voronoi tessellation causes every seam to be at the same distance from the two corresponding image centers. As lens distortions is a radial effect, features that are perpendicular to the seam will be distorted equally on the seam, and therefore will remain aligned regardless of lens distortion.

5 Color Merging in Seams

Changes in image brightness, usually caused by the mechanism of automatic gain control (AGC), cause visible brightness seams in the mosaic between regions covered by different images. These seams should be eliminated in order to get a seamless panorama.

The process of blending the different images into a seamless panorama must smooth all these illumination discontinuities, while preserving image sharpness. A method that fulfills this requirement is described in [4]. In this approach, the images are decomposed into band-pass pyramid levels, and then combined at each band-pass pyramid level. Final reconstruction of the images from the combined band-pass levels give the desired panorama.

6 Examples

Figure 3 shows some panoramic mosaic images created with VideoBrush. More examples can be viewed in "http://www.sarnoff.com/VideoBrush".

7 Concluding Remarks

Manifold Projection enables the fast creation of low-distortion panoramic mosaics under very general camera motions. Implementation under the assumptions of limited change of scale and limited parallax gives unparalleled speed and quality of mosaicing. Future extensions will address the issues of motion parallax, as well as forward motion and zoom which are not addresses in the current scheme.

Bibliography

[1] *ARPA Image Understanding Workshop*, Monterey, California, November 1994. Morgan Kaufmann.

[2] *Fifth International Conference on Computer Vision*, Cambridge, MA, June 1995. IEEE-CS.

Figure 3: Examples of panoramic images using manifold projection. The curved boundary is created by the unstabilized motion of the hand-held camera.

[3] F. Aurenhammer. Voronoi diagrams: A survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, September 1991.

[4] P.J. Burt and E.H. Adelson. A multiresolution spline with application to image mosaics. *ACM Trans. on Graphics*, 2(4):217–236, October 1983.

[5] P.J. Burt and P. Anandan. Image stabilization by registration to a reference mosaic. In *ARPA Image Understanding Workshop* [1],, pages 457–465.

[6] P.J. Burt, R. Hingorani, and R.J. Kolczynski. Mechanisms for isolating component patterns in the sequential analysis of multiple motion. In *Proc. IEEE Workshop on Visual Motion*, pages 187–193, Princeton, NJ, October 1991. IEEE-CS.

[7] Tom R. Halfhill. See you around. *Byte Magazine*, pages 85–90, May 1995.

[8] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P.J. Burt. Real-time scene stabilization and mosaic construction. In *ARPA Image Understanding Workshop* [1], pages 457–465.

[9] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *Fifth International Conference on Computer Vision* [2], pages 605–611.

[10] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In G. Sandini, editor, *Second European Conference on Computer Vision*, pages 282–287, Santa Margherita, Italy, May 1992. Springer.

[11] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 454–460, Seattle, WA, June 1994.

[12] P. Jaillon and A. Montanvert. Image mosaicking applied to three-dimensional surfaces. In *12th International Conference on Pattern Recognition*, pages 253–257, Jerusalem, Israel, October 1994. IEEE-CS.

[13] Arun Krishnan and Narendra Ahuja. Panoramic image acquisition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 379–384, San Fransisco, California, June 1996.

[14] Steve Mann and Rosalind Picard. Virtual bellows: Constructing high quality stills from video. In *First IEEE International Conference on Image Processing*, Austin, Texas, November 1994.

[15] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *SIGGRAPH*, Los Angeles, California, August 1995. ACM.

[16] H.S. Sawhney, S. Ayer, and M. Gorkani. Model-based 2D & 3D dominant motion estimation for mosaicing and video representation. In *Fifth International Conference on Computer Vision* [2], pages 583–590.

MULTI-IMAGE ALIGNMENT

Harpreet S. Sawhney Rakesh Kumar
David Sarnoff Research Center
CN5300, Princeton, NJ 08530
{sawhney,kumar}@sarnoff.com

Abstract

Multiple images of a scene are related through 2D/3D view transformations and linear and non-linear camera transformations. In all the traditional techniques to compute these transformations, especially the ones relying on direct intensity gradients, one image and its coordinate system have been assumed to be ideal and distortion free. In this paper, we present a formulation and an algorithm for *true* multi-image alignment that does not rely on the measurements of a reference image being distortion free. For instance, in the presence of lens distortion, none of the images can be assumed to be ideal. In our formulation, all the images are modeled as intensity measurements represented in their respective coordinate systems, *each* of which is related to an ideal coordinate system through an interior camera transformation and an exterior view transformation. The goal of the accompanying algorithm is to compute an image in the ideal coordinate system while solving for the transformations that relate the ideal system with each of the data images.

Key advantages of the technique presented in this paper are: (i) no reliance on one distortion free image, (ii) ability to register images and compute coordinate transformations even when the multiple images are of an extended scene with no overlap between the first and last frame of the sequence, and (iii) ability to handle linear and non-linear transformations within the same framework.

The new algorithm is evaluated in the context of two applications: (i) correction of lens distortion, and (ii) creation of video mosaics.

1 Introduction

Multiple images of a scene are related through 2D/3D view transformations and linear and non-linear camera transformations. Automatic computation of these transformations is important for applications like image/video mosaicing, structure from motion, and recovery of camera and object motions. Direct methods for simultaneously computing the correspondences between frames and the unknown transformations through alignment have been actively explored in the past few years. Direct estimation has proven to be more practical and robust over more traditional feature correspondence based methods since the direct methods typically use all the data available in images and employ geometric viewpoint and structure constraints in the estimation process. Direct methods have been fruitfully employed in a hierarchical coarse-to-fine optimization framework to estimate 2D parametric transformations [Bergen and others, 1992, Black and Anandan, 1996], 3D view and parallax estimates [Kumar *et al.*, 1994, Sawhney, 1994] both over two and multiple frames [Hanna and Okamoto, 1993], 2D layered and moving object representations [Ayer and Sawhney, 1995, Hsu *et al.*, 1994, Irani *et al.*, 1992], and to create 2D and 3D aligned video mosaics [Irani, 1995, Kumar *et al.*, 1995, Mann and Picard, 1994, Sawhney *et al.*, 1995, Szeliski, 1994].

In all the direct techniques, one image and its coordinate system have been assumed to be ideal and distortion free. In this paper, we present a formulation and an algorithm for *true* multi-image alignment that does not rely on the measurements of a reference image being distortion free. For instance, in the presence of lens distortion, none of the images can be assumed to be ideal. In our formulation, all the images are modeled as intensity measurements represented in their respective coordinate systems, *each* of which is related to an ideal coordinate system through an interior camera transformation and an exterior view transformation. The goal of the ac-

companying algorithm is to compute an image in the ideal coordinate system while solving for the transformations that relate the ideal system with each of the data images. The algorithm is based on a minimum variance estimate of the ideal image that is computed using direct multi-resolution methods.

Key advantages of our technique are: (i) no reliance on one ideal and distortion free image, (ii) ability to register images and compute coordinate transformations even when the multiple images are of an extended scene with no overlap between the first and last frame of the sequence, and (iii) ability to handle linear and non-linear transformations within the same framework.

In Section 2, the formulations of the multi-view variance error function and an iterative solution are presented. Section 3 presents the optimization strategy. Subsequently, we present experimental results for the new algorithm for two applications: (i) correction of lens distortion, and (ii) creation of video mosaics. Finally, in the appendix, some experiments on the validation of our lens distortion model are presented.

2 Formulation

Given images $I_1 \ldots I_N$, the coordinate system of each I_i is represented as a transformed version of an ideal reference coordinate system typically not belonging to any particular image. Therefore, a point $\mathbf{p} = (x, y)$ in the ideal system is related to an observed point $\mathbf{p^i} = (x^i, y^i)$ in the *ith* image through a two-step transformation. In the first step, $\mathbf{p}$ is transformed through a transformation, $\mathbf{A}^i$, which typically is a 3D-to-2D or 2D-to-2D projection transformation, to an undistorted coordinate $\mathbf{p_I^i} = (x_I^i, y_I^i)$. In the second step, $\mathbf{p_I^i}$ is further transformed, typically through a nonlinear camera transformation, γ, to obtain the observed video coordinate $\mathbf{p^i} = (x^i, y^i)$. For simplicity and without loss of generality, γ is assumed to be the same for each image. The functional relationship between a reference coordinate $\mathbf{p}$ and the corresponding video coordinate can be succinctly expressed as:

$$\mathbf{p^i} = \mathbf{p_I^i} + \mathbf{\Gamma}(\mathbf{p_I^i}; \gamma) \qquad \mathbf{p_I^i} = \mathbf{P}(\mathbf{p}; \mathbf{A}^i) \qquad (1)$$

where $\mathbf{P}$ and $\mathbf{\Gamma}$ represent the projection and nonlinear camera tranformations, respectively.

Given the coordinate transformations, intensities at points $\mathbf{p^m}$ in image I_m and at $\mathbf{p^n}$ in image I_n, that transform to the same reference coordinate $\mathbf{p}$, are related through

$$I_m(\mathbf{p^m}; \mathbf{p}, \mathbf{A}^m, \gamma) = I_n(\mathbf{p^n}; \mathbf{p}, \mathbf{A}^n, \gamma). \qquad (2)$$

However, the parameters $\mathbf{A}^1 \ldots \mathbf{A}^N$ and γ are unknown and so is the correspondence between the points of the images. The correspondence between points in various images can be established only through the transformation of the reference coordinates in Equation (1).

In order to compute the correspondences and the unknown parameters simultaneously, we formulate an error function that minimizes the variance in intensities of a set of corresponding points in the images, that map to the same ideal reference coordinate. Formally, the optimization problem is:

$$\min_{\mathbf{A}^1 \ldots \mathbf{A}^N, \gamma} \sum_{\mathbf{p}} \frac{1}{M(\mathbf{p})} \sum_i (I_i(\mathbf{p^i}) - \bar{I}(\mathbf{p}))^2, \qquad (3)$$

where point $\mathbf{p^i}$ in frame i is a transformation of a point $\mathbf{p}$ in the reference coordinate system, $\bar{I}(\mathbf{p})$ is the mean intensity value of all the $\mathbf{p^i}$'s that map to $\mathbf{p}$, and $M(\mathbf{p})$ is a count of all such $\mathbf{p^i}$'s. Therefore, given a point $\mathbf{p}$ in the reference coordinates, each term in the sum in Equation (3) is the variance of all the intensity values at points $\mathbf{p^i}$ that map to point $\mathbf{p}$.

We now develop the multi-image formulation using a parametric plane projective transformation as the scene to image mapping, and lens distortion as the nonlinear camera transformation. This can be specialized and generalized to other parametric (e.g. translation and affine) and quasi-parametric (plane+parallax) models. Recall that the plane projective model with lens distortion captures accurately the image transformations from a real camera undergoing approximately rotations (pan/tilt) and zoom, and also models the other internal camera parameters.

The transformation consists of (for the purposes of this formulation but is not limited to) an 8-parameter plane projective transformation and a 1 or 2 parameter lens distortion transformation. Therefore, Equation (1) can now be written specifically in terms of the transformation parameters as:

$$x_I^i = \frac{a_{11}^i x + a_{12}^i y + a_{13}^i}{a_{31}^i x + a_{32}^i y + a_{33}^i}$$
$$y_I^i = \frac{a_{21}^i x + a_{22}^i y + a_{23}^i}{a_{31}^i x + a_{32}^i y + a_{33}^i}. \qquad (4)$$

where $a_{11} \ldots a_{33}$ are the plane projective parameters with a_{33} set to 1 without loss of generality. (x_I^i, y_I^i) is further transformed non-linearly using the lens distortion to obtain the observed video coordinate (x^i, y^i) through

$$x^i = x_I^i + \gamma_1 (x_I^i - x_C^i) r^2$$
$$y^i = y_I^i + \gamma_1 (y_I^i - y_C^i) r^2 \qquad (5)$$

where $\mathbf{p^i}_C = (x^i_C, y^i_C)$ is the image center for the ith frame, and $r^2 = (x^i_I - x^i_C)^2 + (y^i_I - y^i_C)^2$ is the squared distance of (x^i_I, y^i_I) from the center.

The above equation models only the cubic term of radial lens distortion. For most of the cameras we have experimented with, this is the most significant term. However, the alignment technique presented in this paper can easily be applied to other more general models of lens distortion. For simplicity, it is assumed that each video frame is distorted with the same lens distortion parameter γ_1; not an unreasonable assumption for many real scenarios. It is also assumed that the x and y scale factors for the frame coordinates are the same; otherwise two parameters for lens distortion can easily be specified in the above equation. It is to be noted that if there is no non-linear distortion (as in Equation (5)), then the observed coordinates of one image can be chosen as the reference coordinates. This is a special case of the above formulation.

2.1 Iterative Solution

It is evident from the optimization function in Equation (3) and the transformations in Equations (4)–(5) that the unknown parameters cannot be obtained in closed form. We employ the Levenberg-Marquardt technique for minimizing sum of squares error functions.

In order to apply the LM technique, each term in Equation (3) is linearized. Each term is of the form:

$$E((\mathbf{p^i};\mathbf{p});\mathcal{A},\gamma_1) = \frac{1}{\sqrt{M(\mathbf{p})}}(I_i(\mathbf{p^i}) - \bar{I}(\mathbf{p})), \quad (6)$$

where $\mathcal{A}$ represents the set of all the N unknown $\mathbf{A^i}$'s. Given a solution of the unknown parameters $\mathcal{A}_k, \gamma_{1_k}$ at the kth step in the optimization process, each $E((\mathbf{p^i};\mathbf{p});\mathcal{A},\gamma_1)$ is linearized around this solution as:

$$E((\mathbf{p^i};\mathbf{p});\mathcal{A},\gamma_1) \approx E((\mathbf{p^i};\mathbf{p});\mathcal{A}_k,\gamma_{1_k}) + \nabla E\mid_{\mathcal{A}_k,\gamma_{1_k}} \left[\begin{array}{cc} \delta\mathbf{A}^{1^T} & \ldots \delta\mathbf{A}^{N^T} \delta\gamma_1 \end{array}\right]^T$$
$$= E((\mathbf{p^i};\mathbf{p});\mathcal{A}_k,\gamma_{1_k}) + \nabla E\mid_{\mathcal{A}_k,\gamma_{1_k}} [\delta\mathcal{A}^T \delta\gamma_1]^T] \quad (7)$$

where

$$E((\mathbf{p^i};\mathbf{p});\mathcal{A}_k,\gamma_{1_k}) = \frac{1}{\sqrt{M(\mathbf{p})}} I_i(\mathbf{p^i}(\mathbf{p};\mathbf{A}^i_k,\gamma_{1_k})) - \bar{I}(\mathbf{p^i}(\mathbf{p};\mathbf{A}^i_k,\gamma_{1_k}))$$

The first term on the right hand side in the above equation is the intensity value for image i sampled at location $\mathbf{p^i}$ which is a forward mapping of the corresponding point $\mathbf{p}$ in the reference image with mapping parameters $\mathbf{A}^i_k, \gamma_{1_k}$. Recall that we do not apriori know the correspondences $(\mathbf{p^i}, \mathbf{p})$; these are known only through the mapping parameters. Given that typically only the forward mapping from $\mathbf{p}$ to $\mathbf{p^i}$ is known, $I_i(\mathbf{p^i}(\mathbf{p};\mathbf{A}^i_k,\gamma_{1_k}))$ can be written in terms of a warped image represented in the reference $\mathbf{p}$ coordinates, that is, $I_i(\mathbf{p^i}(\mathbf{p};\mathbf{A}^i_k,\gamma_{1_k})) = I^w_i(\mathbf{p})$. The warped image is created by computing $\mathbf{p^i}$ for image i using $\mathbf{p}$ and the parameters $\mathbf{A}^i_k, \gamma_{1_k}$, and interpolating the known values of the image I_i at integer pixel locations. Therefore, $I^w_i(\mathbf{p})$ represents the current estimate of image i represented in the reference coordinates. Also,

$$\bar{I}(\mathbf{p^i}(\mathbf{p};\mathbf{A}^i_k,\gamma_{1_k})) = \frac{1}{M(\mathbf{p})}\sum_i I^w_i(\mathbf{p}).$$

Therefore, at the kth parameter values,

$$E((\mathbf{p^i};\mathbf{p});\mathcal{A}_k,\gamma_{1_k}) = I^w_i(\mathbf{p}) - \bar{I}^w(\mathbf{p}).$$

The gradient term in the first order approximation of Equation (7) can be written as,

$$\nabla E\mid_{\mathcal{A}_k,\gamma_{1_k}} = \frac{1}{\sqrt{M(\mathbf{p})}}((\nabla I_i(\mathbf{p^i})\nabla\mathbf{p^i})\mid_{\mathbf{p^i}(\mathbf{p};\mathbf{A}^i_k,\gamma_{1_k})} - \frac{1}{M(\mathbf{p})}\sum_i(\nabla I_i(\mathbf{p^i})\nabla\mathbf{p^i})\mid_{\mathbf{p^i}(\mathbf{p};\mathbf{A}^i_k,\gamma_{1_k})}) \quad (8)$$

The gradients of images are in their respective coordinate systems and are 1×2 matrices. Again, the gradient images are represented in the reference coordinate system for particular values of the unknown parameters, $\mathcal{A}_k, \gamma_{1_k}$, through interpolation and warping. The gradients of the ith image's (I_i) coordinate system, $\mathbf{p^i}$, are with respect to the unknown parameters, $\mathbf{A}_i, \gamma_1$, evaluated at the current values $\mathbf{A}^i_k, \gamma_{1_k}$. Each of these is a $2 \times (N \times M + 1)$ matrix where M is the dimension of each unknown parameter vector $\mathbf{A}_i$, N is the number of images, and 1 accounts for the unknown scalar γ_1.

The gradients of the image coordinates can be expressed as:

$$\nabla_{(\mathbf{A}^i,\gamma_1)}\mathbf{p^i} = \nabla_{\mathbf{A}^i}\,\mathbf{p^i_I} + \nabla_{(\mathbf{A}^i,\gamma_1)}\mathbf{\Gamma}(\mathbf{p^i_I};\gamma_1)$$

The gradients are separated into the ones with respect to $\mathbf{A}^i$ and γ_1. From Equation (1),

$$\nabla_{\mathbf{A}^i}\mathbf{p^i} = ((1+\gamma_1 r^2)\mathbf{I_2} + 2\gamma_1[\mathbf{p^i_I} - \mathbf{p^i}_C][\mathbf{p^i_I} - \mathbf{p^i}_C]^T)\nabla\mid_{\mathbf{A}^i}\mathbf{p^i_I},$$

where $\mathbf{I_2}$ is the 2×2 identity matrix.

Using the augmented vector $\mathbf{p_a} = \left[\begin{array}{cc} \mathbf{p} & 1 \end{array}\right]^T$,

$$\nabla_{\mathbf{A}^i}\mathbf{p^i_I} = \left[\begin{array}{ccc} \mathbf{h_1} & \mathbf{0_3} & \mathbf{h_2} \\ \mathbf{0_3} & \mathbf{h_1} & \mathbf{h_3} \end{array}\right]$$

where

$$\begin{aligned}
\mathbf{0_3} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \\
\mathbf{h_1} &= \begin{bmatrix} \frac{1}{\mathbf{A_3^{i\,T}p_a}}x & \frac{1}{\mathbf{A_3^{i\,T}p_a}}y & \frac{1}{\mathbf{A_3^{i\,T}p_a}}1 \end{bmatrix} \\
\mathbf{h_2} &= \begin{bmatrix} -\frac{1}{\mathbf{A_3^{i\,T}p_a}}x_I^i x & -\frac{1}{\mathbf{A_3^{i\,T}p_a}}x_I^i y \end{bmatrix} \\
\mathbf{h_3} &= \begin{bmatrix} -\frac{1}{\mathbf{A_3^{i\,T}p_a}}y_I^i x & -\frac{1}{\mathbf{A_3^{i\,T}p_a}}y_I^i y \end{bmatrix} \\
\mathbf{A}^i &= \begin{bmatrix} \mathbf{A_1^{i\,T}} & \mathbf{A_2^{i\,T}} & \mathbf{A_3^{i\,T}} \end{bmatrix}^T.
\end{aligned}$$

Furthermore, $\nabla_{\gamma_1}\mathbf{p^i} = [\mathbf{p_I^i} - \mathbf{p^i}_C]r^2$.

Let $\mathbf{g}^i = \nabla I_i(\mathbf{p^i})\nabla_{\mathbf{A}^i}\mathbf{p^i}$ be a $1 \times M$ matrix, and $g^i = \nabla I_i(\mathbf{p^i})\nabla_{\gamma_1}\mathbf{p^i}$ be a scalar. Also let $\mathbf{G(p)} = [\mathbf{g}^1 \ldots \mathbf{g}^i \ldots \mathbf{g}^N]$ be the $1 \times M * N$ matrix of all the $\mathbf{g}^i$'s, and $g = \sum_i g^i$. Then ∇E of Equation (8) can be written as:

$$\nabla E = \frac{1}{\sqrt{M(\mathbf{p})}}[0\,0 \ldots \mathbf{g}^i\,0 \ldots 0\,g^i] - \frac{1}{M(\mathbf{p})}[\mathbf{G(p)}\;g].$$

Each iteration solves the following linear sum of squares problem using LM:

$$\sum_{\mathbf{p}}\sum_{i}(E(\mathbf{p^i};\mathbf{p}) + \nabla E(\mathbf{p^i};\mathbf{p})\begin{bmatrix} \delta\mathcal{A} \\ \delta\gamma_1 \end{bmatrix})^2.$$

For each point $\mathbf{p}$ in the reference coordinates, all the images that contain a point that maps to $\mathbf{p}$ contribute an equation to the system of equations corresponding to the above problem. LM iterations look for a solution that results in a reduction of the error function by making the Hessian diagonally dominant progressively, if the original system leads to an increase in the error value. In order to obtain a well-conditioned system of equations, the unknown parameters are scaled appropriately so that the Hessian remains well conditioned.

3 Minimization Strategy

In order to handle a wide range of motion between frames, and to efficiently compute large number of parameters through frame alignment, we adopt an optimization strategy that uses (i) progressively complex models of motion and (ii) coarse-to-fine tracking of the model parameters.

Progressive Complexity

In order to solve for a large number of parameters (typically, $8N+1$ for $N+1$ frames with their respective plane projective transformations and a common lens distortion parameter) efficiently, we have found empirically that the use of models with a progressive increase in complexity helps tremendously. Spurious local minima are avoided and the number of iterations required is considerably reduced.

The progressive complexity strategy is to divide the optimization process into a sequence of steps. At each step, an increasingly higher parametric order motion model is inserted in Equation (3) and the subsequent error function is minimized. The results from the previous step are used as an initial estimate for the next step. The unknown projective parameters can be decomposed into the following hierarchy for estimation:

1. 2D Translation, 2 unknown parameters,a_{13}^i, a_{23}^i, for each frame (Equation (4)).

   We first solve for only the translation parameters within a region of interest which is limited to an inner central square, typically $\frac{1}{3}$ of the input images along each dimension. Pixels at the inner central square suffer from little lens distortion as compared to pixels in the outer boundaries.

2. Affine, 6 unknown parameters $a_{11}^i, a_{12}^i, a_{13}^i, a_{21}^i, a_{22}^i, a_{23}^i$.

   The initial translation is used to solve for affine parameters. The region of interest is expanded a little (to a dimension of $\frac{2}{3}$ of the image), but still does not cover the whole image.

3. Projective, 8[1] parameters plus the global lens distortion parameters as in Equations (4) and (5).

   Finally, the affine parameters are used as an initial estimate for computing the projective and the lens distortion parameters simultaneously. In this step, the error function is optimized over the entire image.

In some situations step 2 may be skipped.

Coarse to Fine Minimization

In addition to the progressive complexity strategy, in order to align frames with displacements in tens of pixels, optimization over coarse-to-fine levels of a gaussian/laplacian pyramid is necessary. The parameters are first estimated at the coarse level of the pyramid and the results from this level are used as an initial estimate for the next finer level of the pyramid.

Typically the two strategies are combined. At the higher levels of the pyramid, only the low order models are computed. The results from these are used

[1]Note, the projective transformation can also be modeled by 9 parameters, with the constraint that the RMS value for the 9 parameters is equal to 1.

as an initial estimate for solving the higher order models at the fine levels of the pyramid.

4 Experiments with Lens Distortion Correction

One of the applications for multi-image registration is video mosaics using off-the-shelf inexpensive PC cameras. Severe lens distortion is a common occurrence in most of these cameras. In order to create high quality mosaics using these cameras, it is necessary to correct for the distortion. Our algorithm may be used for this purpose either to compute the lens distortion parameter in an initializing phase in which only a few frames are used or along with the computation of the alignment parameters for each frame.

We first show the results of computing the lens distortion parameters from a few frames. In principle, two frames should be sufficient to solve for the view transformation and lens distortion. However, we have observed that often two frames lead to local minimum solutions that can be avoided by using three frames.

Room Sequence

The first experiment is on a *room* sequence. The sequence was acquired through a hand held inexpensive Toshiba desktop CCD camera. The effort was to capture a sequence of the complete room (about 180 degrees) through two roughly panning swipes of the camera.

The multi-frame registration algorithm was applied on three frames, shown in figure 1, with a plane projective and lens distortion model. The three aligned frames are shown in the undistorted coordinate system of the middle frame in Figure 2. The frames are shown in a frame bigger than the original to show the full extent of the warping with the projective and lens distortion parameters. Figure 3 shows the differences with respect to the reference frame before and after alignment, in the original size of the frames.

Document Sequence

Three images of a document scanned using an inexpensive Visual Labs. "gooseneck" camera are shown in Figure 5. The 3D motion used to acquire the images is essentially a global y-axis translation. Since the paper is flat, the image motion ideally would be described by a global 2D motion. However, from the figure, it can be noted that there is significant radial lens distortion in the images. Figure 6 shows the input images warped by the computed lens distortion and global projective parameters. As can be noted from the images in the figure, the straight lines corresponding to page margins and sentence boundaries appear quite straight, showing effective compensation for lens distortion.

5 Distortion Corrected Video Mosaics

The algorithm demonstrated above may be applied over multiple frames simultaneously to register all of them in a given undistorted reference frame. Alternatively, to avoid solving for a large number of parameters simultaneously, a seed set of frames may be used to compute their view transformations and the lens distortion. Subsequently, the lens distortion is applied as a pre-processing steps to the rest of the frames and only the projective parameters are solved for to align new frames with already aligned ones. An interesting issue in creating such multi-frame video mosaics is whether frames should be registered to their neighbors and subsequently assimilated into a single coordinate system, or a given frame should be aligned to the current mosaic. This issue is extremely important when mosaics of extended scenes are created by panning and tilting the camera and frames contain views of the same scene patches *may not be temporally contiguous.* For instance, the *room* sequence, was captured using two panning scans which were overlapping. Due to the constraints on space in this presentation, we are unable to go into a detailed demonstration of the comparisons between the two approaches. Almost all existing methods have used parameters computed by consecutive frame alignment to create mosaics. We show a mosaic of 8 frames of the *room* scene that was constructed using lens distortion correction applied to each frame and through registration of frames to an evolving mosaic. Only the final result is shown in Figure 4.

A mosaic built using the computed transformations on the document images is shown in Figure 7.

Appendix

A Evaluation of the Lens Distortion Model

We now present preliminary results on the validity of the lens distortion model and our multi-frame parameter estimation technique. Experimental results on the quantitative accuracy achieved in point localization by the techniques presented in this paper

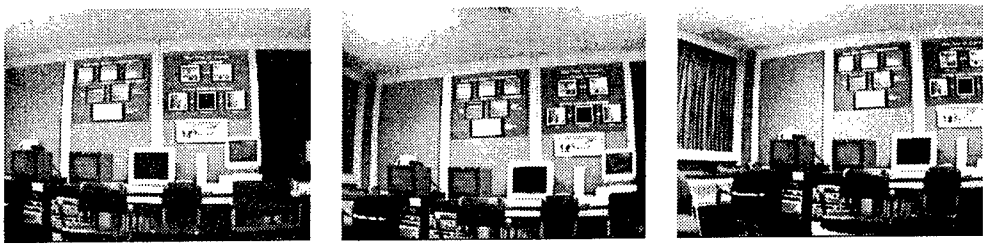


Figure 1: Three frames from *room* sequence through a PC camera with severe lens distortion.

Figure 2: The three frames, registered with the multi-frame plane projective plus lens distortion model, shown as complete warped frames in the coordinate system of the undistorted reference frame.

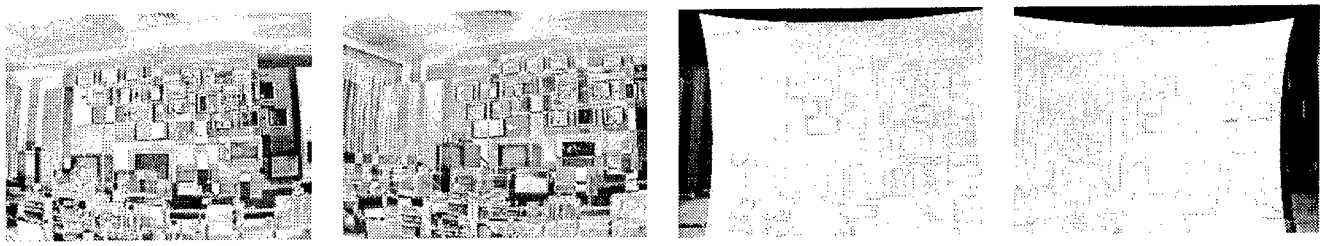


Figure 3: Left two: Difference images for frame differences between the first and the third frames with respect to the second for the *room* sequence. *White denotes low differences and black high differences.* **Right two:** Differences after multi-frame alignment.

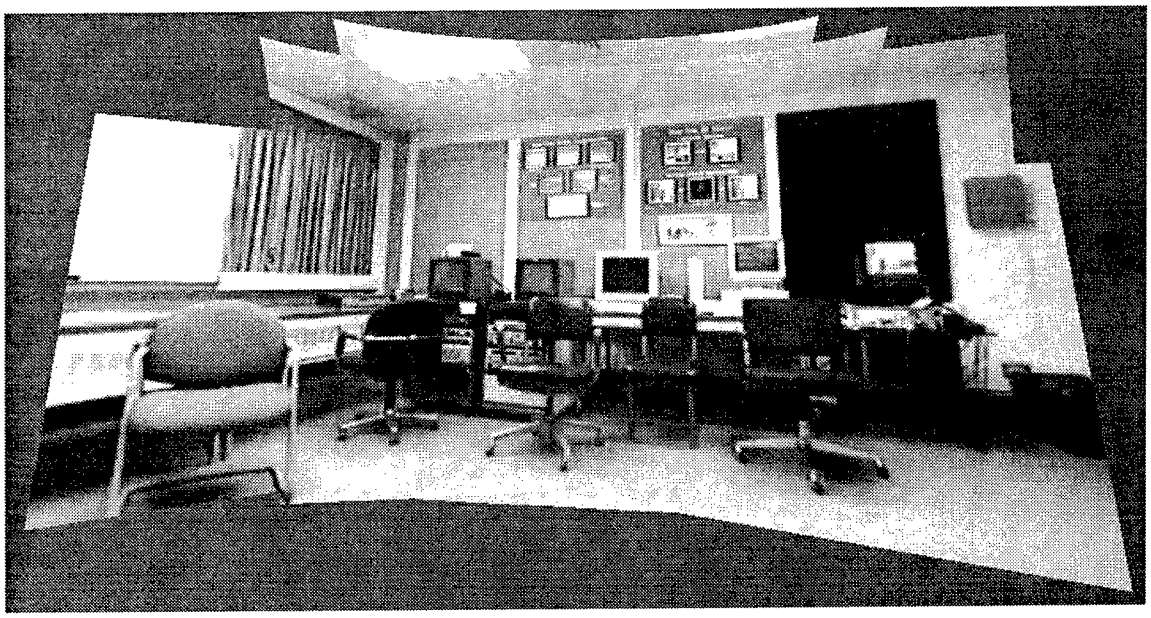


Figure 4: Video mosaic with frame-to-mosaic alignment for the *room* video.

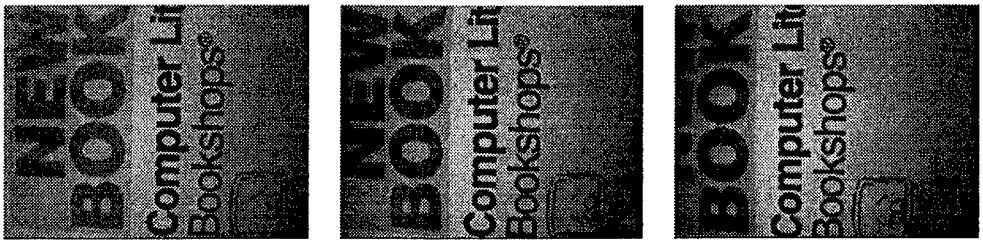


Figure 5: Three frames of a document with severe lens distortion.

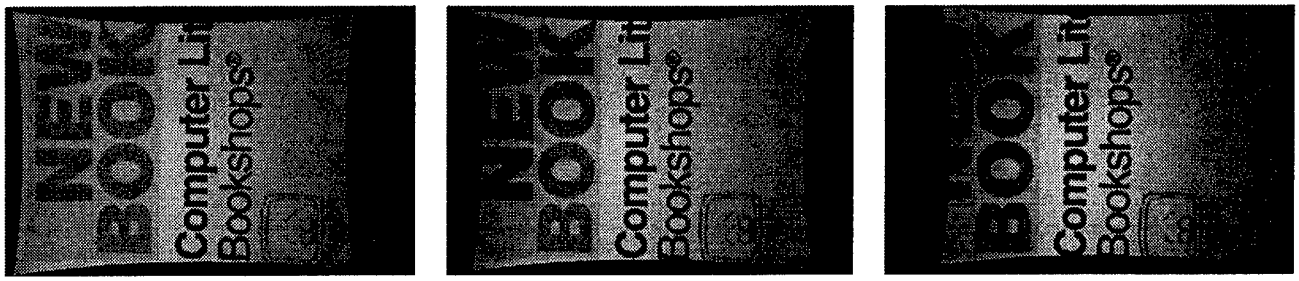


Figure 6: Warped document images after compensation for global projective transformation and radial lens distortion.

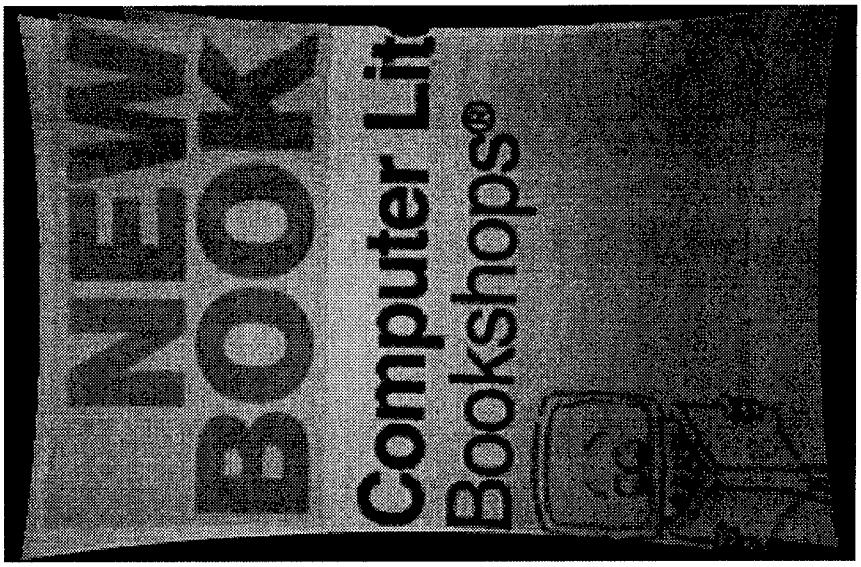


Figure 7: Document mosaic after compensation for lens distortion and projective transformation.

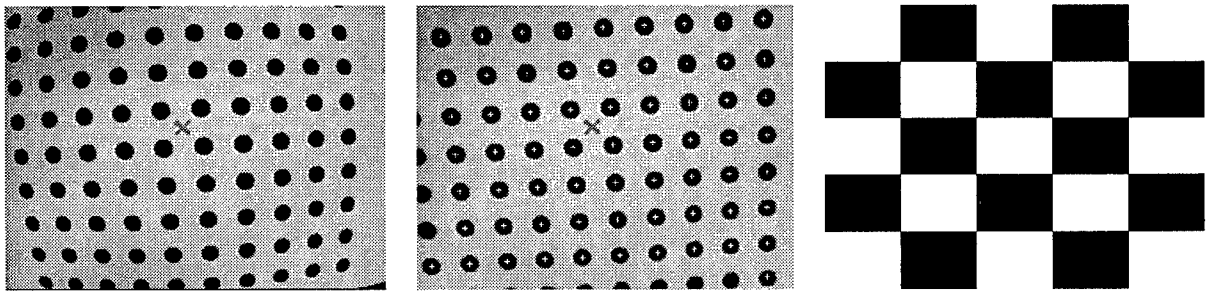


Figure 8: **Left**: One image of a grid of circles showing lens distortion. **Middle**: The image after warping with projective+lens distortion parameters. (*Centers of circles have been marked with a* +). **Right**: The mask used for evaluation of prediction errors based on estimated parameters.

are reported. In contrast with other calibration techniques, that rely on comparing an image of a calibration object/pattern with the true pattern, since our technique relies only on captured images, the evaluation technique also follows this framework. A pattern of uniform sized black circles on a white background was chosen as a test pattern. The uniform sized circles were chosen so that locating their centers and using these to compute point localization errors would be relatively easy. Using a PC camera, a number of images of this test pattern were captured by moving the camera. Subsequently, two tests were performed on the images.

First, 2 or 3 frames of the captured images (one of the images is shown in Figure 8) were used to align the *full* images using four different models: (i) 2D affine, (ii) plane projective, (iii) affine with lens distortion, and (iv) plane projective with lens distortion. Each of the models was used separately to align the images using the method described earlier. Each of the models was plugged in in the optimization of Equation (3). Figure 8 shows one frame after warping with the projective+lens distortion alignment parameters. After alignment, a template of one circular pattern was used to locate to sub-pixel accuracy the centers of all the circular patterns in each of the aligned images. The RMS error between corresponding points is reported as the achievable accuracy with the four models used. These results are reported in Table 1 under the $64pts.$ column.

The second experiment reports the results of predictability of points using the projection and distortion models. Instead of using the whole images for alignment, a checker board binary mask, shown in Figure 8, was used to compute the parameters. Only the image data that corresponds to the white areas of the mask is used for parameter computation; the black areas are ignored. Again, after alignment with the four models with the mask, results are reported for point localization separately for points that lie in the white areas (and hence participated in the estimation process), and for points that lie in the black areas (those that are predicted by the computed parameters but did not participate in the estimation process). These results are reported in Table 1 under the two $32pts.$ columns, one each for points that were used for estimation and the points that were predicted.

In the third evaluation experiment, we report the point location estimation errors for various values of the image center around the nominal image center which is $(160, 120)$ for the images used in the experiment. Note that for each parameter estimation run the image center was kept fixed but was varied between different runs. The projective+lens distortion model was used to align the three grid images using our multi-frame alignment method with different but fixed values of the image center. Table 2 reports the RMS errors for points between the warped reference frame (frame 1) and the other two frames, 0 and 2, for various values of the center. The warped frame represents the predicted image using the computed parameters. The best estimation errors occur for values $(160, 120)$ and $(170, 110)$.

Table 1: Estimation and prediction errors of points for various models. Optical center assumed to be at 160, 120 for the images of size 320, 240. Second column shows RMS errors when all points are used in parameter estimation. Third and fourth columns show the RMS errors for half the points that are used in parameter estimation, and the other half that are only predicted.

Model Type	Estimated RMS error	Estimated RMS error	Predicted RMS error
No. of pts.	64 pts.	32 pts.	32 pts.
	pixels	pixels	pixels
Affn.	1.36	1.17	1.56
Proj.	0.67	0.64	0.72
Affn.+LD	0.60	0.57	0.46
Proj.+LD	0.26	0.39	0.34

In order to estimate both the center and the other parameters automatically, the two center parameters could also be unknowns in the estimation procedure. Alternatively, a search for the best center around the nominal one may be adequate. Note that in order to be able to estimate the appropriate center also automatically, we have to be able to find the best estimation error for the minimum variance estimate and not the point correspondence errors. The results reported here for point correspondences are preliminary. Further work will lead to a better understanding of the relation between the center and the minimum variance estimation error.

Acknowledgments

The work reported here was funded in part by the National Information Display Laboratory, Princeton, NJ. Our thanks to Jane Asmuth for her help in the calibration evaluation experiments.

References

[Ayer and Sawhney, 1995] S. Ayer and H. S. Sawhney. Layered representation of motion video using

Table 2: RMS error of points for optical center positions using projective model with lens distortion. Columns 3 and 4 show the RMS errors, with different centers, for points in frames 0 and 2 with frame 1 chosen as the reference.

Optical Center		Frame 0-1 RMS error	Frame 1-2 RMS error
X	Y	64 pts.	64 pts.
pixel	pixel	pixels	pixels
160	120	0.261	0.289
155	120	0.277	0.337
165	120	0.277	0.307
160	115	0.273	0.297
160	125	0.355	0.310
170	110	0.292	0.242
170	130	0.387	0.404
150	130	0.320	0.358
150	110	0.420	0.430
160	115	0.273	0.297
180	100	0.363	0.320
190	90	0.344	0.353
200	80	0.395	0.418

robust maximum-likelihood estimation of mixture models and MDL encoding. In *ICCV*, 1995.

[Bergen and others, 1992] J. R. Bergen et al. Hierarchical model–based motion estimation. In *2nd ECCV*, pages 237–252, 1992.

[Black and Anandan, 1996] M. J. Black and P. Anandan. The robust estimation of multiple motions: Affine and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

[Hanna and Okamoto, 1993] K.J. Hanna and N. Okamoto. Combining stereo and motion analysis for direct estimation of scene structure. In *International Conference on Computer Vision*, pages 357–365, Berlin, May 1993.

[Hsu *et al.*, 1994] S. Hsu, P. Anandan, and S. Peleg. Accurate computation of optical flow by using layered motion representation. In *ICPR*, pages 743–746, Jerusalem, Israel, Oct. 1994.

[Irani *et al.*, 1992] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *ECCV*, pages 282–287, Santa Margherita, Italy, May 1992.

[Irani, 1995] Michal Irani. Applications of image mosaics. In *International Conference on Computer Vision*, Cambridge, MA, November 1995.

[Kumar *et al.*, 1994] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: A parallax based approach. In *ICPR*, pages 685–688, 1994.

[Kumar *et al.*, 1995] R. Kumar, P. Anandan, M. Irani, et al. Representation of scenes from collection of images. In *Proc. IEEE Wkshp. on Representation of Visual Scenes*, 1995.

[Mann and Picard, 1994] S. Mann and R. W. Picard. Virtual bellows: Constructing high quality stills from video. In *ICIP*, 1994.

[Sawhney *et al.*, 1995] H. S. Sawhney, S. Ayer, and M. Gorkani. Model-based 2D&3D dominant motion estimation for mosaicing and video representation. In *ICCV*, 1995.

[Sawhney, 1994] H. S. Sawhney. Simplifying motion and structure analysis using planar parallax and image warping. In *Proc. Intl. Conf. on Pattern Recognition*, pages A403–A408, 1994.

[Szeliski, 1994] R. Szeliski. Image mosaicing for tele-reality applications. In *IEEE Wkshp. on Applications of Computer Vision*, pages 44–53, 1994.

Horizon Line Matching for Orientation Correction Using a Messy Genetic Algorithm *

Karthik Balasubramaniam **J. Ross Beveridge**
Christopher E. Lesher **Christopher Graves**
Colorado State University
balasub/ross/lesher/gravesc@cs.colostate.edu

Abstract

A robust and general line-matching system is used to match horizon lines extracted from a terrain map to those extracted from CCD imagery. Based upon these matches, uncertainty in camera pointing angle is reduced from several degrees to less than a degree. Results are presented for two vehicle locations using imagery collected by the Unmanned Ground Vehicle program at the Lockheed-Martin Demo C test site. A new heuristic matching technique based upon a Messy Genetic Algorithm is used to obtain the optimal horizon matches.

1 Introduction

In [Beveridge *et al.*, 1996], we demonstrated the feasibility of matching horizon lines extracted from CCD imagery to horizons extracted from rendered terrain maps. The practical application is to automate a process of precisely aligning terrain maps to ground-looking imagery. It was found during the Unmanned Ground Vehicle (UGV) program that even when using an inertial guidance system to track vehicle orientation, errors of up to 1 to 2 degrees in pointing angle were common [Rimey and Hougen, 1995].

Here we extend the work presented in [Beveridge *et al.*, 1996] by testing how well 3D camera orientation can be corrected using automatically matched horizons. We also introduce a new variant upon our past optimal matching work [Beveridge *et al.*, 1990; Beveridge, 1993; Beveridge, 1997] which uses a Messy Genetic Algorithm to control search. Qualitatively speaking, we find the Messy Genetic Algorithm (MGA) performs better than our past local search techniques on the horizon problems; a quantitative comparison of the MGA and local search appears in [Whitley *et al.*, 1997].

The procedure for correcting camera orientation is as follows:

1. Render the 3D terrain model using an estimate of the vehicle pointing angle.
2. Extract the horizon lines from the rendered terrain and the CCD imagery.
3. Use the MGA to optimally match the two sets of horizon lines.
4. Use the matched features to compute the orientation correction.

Section 2 covers terrain rendering while Section 3 describes the extraction of line segments representing the horizon. Section 4 describes the process of finding the optimal 2D horizon matches and Section 5 explains the process of correcting 3D camera pointing angle based

This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) Image Understanding Program under contract 96-14-112 monitored by the Army Topographic Engineering Laboratory (TEC), contracts DAAH04-93-G-422 and DAAH04-95-1-0447, monitored by the U. S. Army Research Office as well as by the National Science Foundation under grant IRI-9503366.

upon these 2D horizon matches. Section 6 describes our experiment design for testing how well camera orientation is corrected and Section 7 presents the results of this experiment.

2 Terrain Rendering

The digital elevation map (DEM) for the terrain covered by our experiments, was obtained during the UGV RSTA program [Rimey, 1995]. We use imagery collected at the Lockheed Martin site at Denver, Colorado. The imagery was captured using a color CCD camera mounted on the UGV. Images for various vehicle positions are available, along with ground truth. Ground truth provides the vehicle position, and an estimate of vehicle pointing angle, based on positions of gimbal targets. These targets are markers on the test site. At each selected UGV position, the pan and tilt rotations needed to center the camera on these targets, gives us an estimate of the pointing angle.

The terrain rendering system, which has been developed using Open-GL, renders the terrain given the vehicle position and pointing angle. The field of view used is 11.11° horizontal, and 7.89° vertical. However, as is evident from a comparison of captured and rendered images, this is not very accurate. A simple lighting model is used, since our interest lies only in the horizon lines. These horizon lines are extracted from a binary thresholded version of the rendered image.

3 Extracting Horizon Lines

The Burns' algorithm [Burns *et al.*, 1986] is used to extract lines from the CCD as well as the rendered imagery.

The presence of high frequency texture in the CCD imagery necessitates smoothing prior to extraction. A 5×5 averaging kernel is used. A problem faced here is that the lines extracted from the CCD imagery are often highly fragmented. We attempt to restrict noise by enforcing a minimum line length constraint, on extracted lines. While we have not developed any domain specific heuristics to improve line extraction for this task, past work [Thompson *et al.*, 1993; Sutherland and Thompson, 1994] suggests that domain specific feature extraction can be useful.

For the terrain map horizons, the thresholded images are directly available from the rendering routine. Figure 3 shows the lines extracted from the CCD image (Figure 1). Figure 4 shows the lines extracted from the rendered terrain image shown in Figure 2.

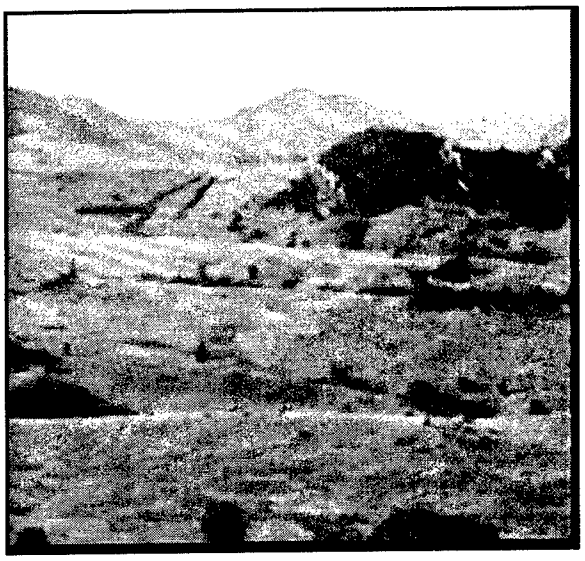


Figure 1: Actual CCD Imagery

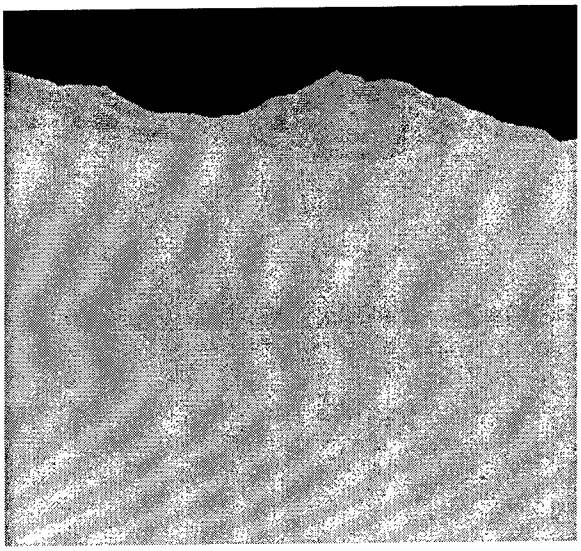


Figure 2: Terrain Map Rendering

Figure 3: Extracted Lines from CCD Image

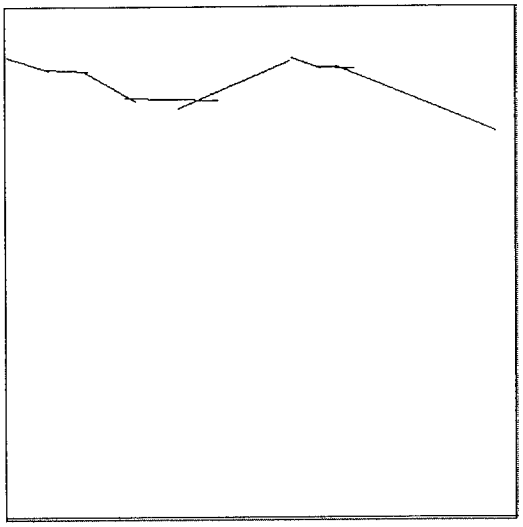


Figure 4: Extracted Lines from Rendered Image

4 2D Line Matching

Line matching determines the correspondence mapping (many-to-many) between a set of model line segments M and data line segments D that minimizes a *match error function.* In the case of horizon line matching, the data line segments are the horizon lines extracted from the actual CCD image. The model line segments are the horizon lines extracted from the rendered image which represents the image that ought to have resulted for the current estimate of pointing angle.

The match error is formulated as the sum of fit and $omission$ errors. The fit error indicates how closely the model fits the data. This value is computed using the *integrated squared perpendicular distance* between corresponding model and data line segments. The omission error measures the extent to which the model line segments are covered by the data. Match error may be written as:

$$E_{match} = \frac{1}{\sigma^2} E_{fit} + E_{omission} \qquad (1)$$

The weighting coefficient σ controls the relative importance of the two error components. We have presented and used E_{match} in several previous works [Beveridge *et al.*, 1990; Beveridge, 1993; Beveridge, 1997] and refer interested readers to these sources for additional detail.

The search space for the matching problem consists of the power set C of all pairs S drawn from M and D. Thus,

$$S \subset M \times D \qquad C = 2^S \qquad (2)$$

The goal of matching is to find the optimal match $c^* \in C$ such that

$$E_{match}(c^*) \leq E_{match}(c) \quad \forall c \in C \qquad (3)$$

In our previous paper [Beveridge *et al.*, 1996] on matching horizon lines, we reported results using an algorithm called subset-convergent local search [Beveridge *et al.*, 1996] to find c^*. Since that time, we've continued to study the problem of horizon line matching and, in particular, have selected two vehicle locations and associated datasets for testing. On these cases, we've found subset-convergent local search to not perform as well as a newly developed technique based upon a Messy Genetic Algorithm.

4.1 Matching with a Messy GA

Messy Genetic Algorithms [Goldberg *et al.*, 1989] differ from normal Genetic Algorithms in that they allow variable-length strings that may be under-specified or over-specified with respect to the problem being solved. For matching geometric models, this means they can operate

over partial matches and thereby piece together larger and better matches.

A Messy Genetic Algorithm typically has three phases:

1. Initialization.
2. Primordial Phase.
3. Juxtapositional phase.

Spatially proximal triples of line segments are used to initialize our MGA, where spatially proximal triples are defines as follows. For each model line $m_i \in M$, the closest two neighbors m_{i1} and m_{i2} as defined by Euclidean distance δ are

$$\delta(m_i, m_{i1}) \leq \delta(m_i, m_k) \ \forall \ m_k \in M - \{m_i\}$$
$$\delta(m_i, m_{i2}) \leq \delta(m_i, m_k) \ \forall \ m_k \in M - \{m_i, m_{i1}\}$$

Similarly, for each data line segment $d_j \in D$ the analogous nearest neighbors are d_{j1} and d_{j2}.

Given a matching problem between M and D, each pair of segments $(m_i, d_j) \in S$ form two spatially proximate triples f_1 and f_2:

$$\begin{aligned} f_1 &= ((m_i, d_j), (m_{i1}, d_{j1}), (m_{i2}, d_{j2})) \\ f_2 &= ((m_i, d_j), (m_{i1}, d_{j2}), (m_{i2}, d_{j1})) \end{aligned} \tag{4}$$

Since each of the n pairs of model and data segments in S leads to 2 triples, there are $2n$ spatially proximate triples.

The modified initialization phase creates $2n$ triples to seed the initial population of the genetic algorithm. Then, in the primordial phase, the match error E_{match} is computed for each of the $2n$ triples. The triples are then sorted, and some fraction of the best form the initial population. In the experiments presented here, the top 50% of triples are used.

During Juxtapostion, selection is used together with two operators: cut and splice. Cut 'cuts' the chromosome at random position. Splice 'attaches' two cut chromosomes together. These two operators are the equivalents of crossover in a traditional GA. In our matching problem, a chromosome h is a variable length set of pairs $h \subset S$ representing a match between model and data segments. Thus, using the cut and splice operators, the MGA will progressively assemble better and better matches. All newly created matches are ranked by E_{match}.

At some point, recombination will typically construct enough of the match for local search to easily and quickly fill out the rest. For this reason, a pass of steepest-descent local search [Beveridge, 1993; Beveridge, 1997] is periodically applied to individuals from the population. The frequency with which local search is run increases as population size decreases.

To help drive the Messy Genetic Algorithm to a solution, every three generations the least fit individual in the population is dropped and the population size correspondingly shrinks by one. Every $f = \frac{p}{2}$ generations, an individual is selected from the population and local search is run using the selected match as an initial state. If the result is better than the worst currently in the population, then it is inserted back into the population.

5 Orientation Correction

Given the two sets of line segments, the MGA determines the transformation of the model corresponding to the best match. For our purposes, we only need the transformation between the horizon as it appears in the terrain rendered image and the horizon is it appears in the CCD image.

This 2D transformation is returned by the MGA when it finds the best match. The transformation is specified by parameters:

$$s, \quad \Phi = (\phi_1, \phi_2), \quad T = (T_1, T_2)$$

representing scale, rotation, and translation respectively. The transformation is effected as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \phi_1 & -\phi_2 \\ \phi_2 & \phi_1 \end{bmatrix} \left(s \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} T_2 \\ T_1 \end{bmatrix} \right) \tag{5}$$

This transformation maps a point (x, y) in the rendered image into a point (x', y') in the CCD image. This transformation lets us create a set of paired points, with each pair containing a point coordinate in the terrain map image and the corresponding coordinate in the CCD image. In principle we could do this with any set of points lying on the horizon. In practice we use

the endpoints of the segments extracted from the rendered terrain image.

This 2D mapping between points in the CCD and rendered image can be used to create a set of corresponding 3D points by backprojecting each 2D image point into the scene. Since we are concerned only with camera rotation, the exact depth we choose for back projecting the image points does not matter.

Each 3D point derived from the CCD image, $(x^{'}, y^{'}, z^{'})$, is now paired with a 3D point derived from the terrain rendered image, (x, y, z). The rotation R which corrects the orientation between terrain and CCD images should satisfy

$$\begin{bmatrix} x^{'} \\ y^{'} \\ z^{'} \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{6}$$

This rotation is the orientation correction. In other words, R will correct for a mis-alignment between the orientation of the camera and the terrain map.

We solve for the 3D rotation which aligns our two sets of points using a subset of Horn's *absolute orientation* method [Horn, 1987]. Let $\vec{r}_d$ denote the set of model points transformed to match the data points, and $\vec{r}_m$, the model points.

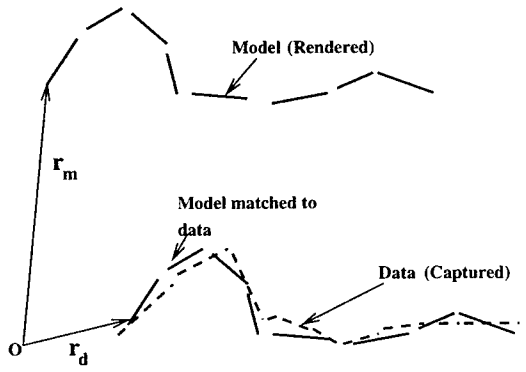


Figure 5: Horizon Line Matching

We need to find the rotation R that maximizes

$$\sum_{i=1}^{n} \vec{r}_m \cdot R(\vec{r}_d)$$

This may be alternatively formulated as finding the quaternion $\dot{q}$ that maximizes.

$$\sum_{i=1}^{n} (\dot{q}\dot{r}_d\dot{q}^*) \cdot \dot{r}_m$$

The correction rotation, R, is used to obtain an estimate of the true pointing angle.

The field of view estimate originally supplied for the Lockheed-Martin CCD camera is not accurate. This is evident visually if one carefully compares Figures 1 and 2. The optimal 2D matching does recover a scale correction through scaling the terrain map horizon to match the image. We can use this scale term from the match to correct the field of view as follows:

$$\theta_{corrected} = 2tan^{-1}\left(\frac{tan\theta_{original}/2}{s}\right) \tag{7}$$

6 Experiments

Experiments were run for two different positions of the UGV (Scenarios 13 and 10, [Rimey, 1995]) on the test site. The horizons in these CCD images are reasonably distinctive as required by this technique. The gimbal targets were used to obtain ground truth estimates of the pointing angle for each scenario. Perturbations applied to this pointing angle yielded sets of test cases. The pan perturbation (Δpan) values used were $-2°, 0°$, and $+2°$, the tilt perturbation ($\Delta tilt$) values were $-1°, 0°$, and $+2°$, and the roll perturbation ($\Delta roll$) values were $-5°, 0°$, and $+5°$. Thus, there are a total of 27 test cases for each UGV position under consideration.

The conventions followed are that positive pans rotate the camera to the right, positive tilts rotate the camera upwards, and positive rolls rotate the camera clockwise. The terrain rendering, horizon line matching, and orientation correction operations were performed for each of the 54 cases, and the residual rotations were obtained. The matching process involved 100 trials of the Messy Genetic search algorithm. It is worthwhile to run multiple trials, since we are using a stochastic search algorithm.

7 Results

We present results first for the optimal matching part of our experiment and then for the 3D orientation corrections derived from these matches.

7.1 Results of Matching

The implementation of the matching system running on a Sparc 20 workstation takes roughly 1 hour to complete 100 independent trials of the MGA on Scenario 13. Scenario 10 takes only about 15 minutes to run 100 trials. Scenario 13 has some of the largest search spaces for which the matching system has been used.

Table 7.1 contains summaries of the matching trials for Scenarios 13 and 10. The number of possible model-data pairings is denoted by n. Average run-time per trial is given by r_{pt}. P_s denotes the probability of finding the optimal match on any given independent run of the Messy Genetic Algorithm. The probability of seeing the optimal match at least once, over t independent trials, is

$$Q_s = 1 - (1 - P_s)^t \tag{8}$$

Therefore, the number of trials t_s required to find the optimal match with probability Q_s is given by:

$$t_s = \lceil \log_{1-P_s}(1 - Q_s) \rceil \tag{9}$$

The expected run-time is given by

$$r_s = t_s r_{pt} \tag{10}$$

The computed values of t_s, and expected run-times (for the machine used), are included in Table 2. We use a confidence level of 95% ($Q_s = 0.95$). As is evident from the table, Scenario 13 is a harder problem than Scenario 10. This arises from the fact that there is more noise in the lines extracted from the CCD image, and more ambiguity as well.

7.2 Results of Orientation Correction

The arrays of images in Figure 6 depict pre-correction and post-correction orientations for

	n	r_{pt}	E_{Match}	P_s
average	1085	72	0.416	57
minimum	456	19	0.253	1
maximum	2083	158	0.623	100
median	1140	64	0.411	49
σ	429	36	0.075	40

(a) Scenario 13

	n	r_{pt}	E_{Match}	P_s
average	364	10	0.265	97
minimum	247	7	0.169	56
maximum	469	14	0.358	100
median	371	10	0.266	100
σ	62	2	0.046	9

(b) Scenario 10

Table 1: Summary of Matching Results.

	t_s	r_s (seconds)
Scenario 13	3	254
Scenario 10	1	8

Table 2: Predicted Run-times for 95% Confidence

Scenarios 13 and 10 as manifested in terrain renderings from the vehicle position and orientation. These figures are provide some intuition for how much these changes in viewing angles change the relative placement of the horizon in the image.

The scatter plots in Figure 7.2 depict orientation errors, for Scenarios 13 (plots (a) through (c)) and 10 (plots (d) through (f)). The x-axis denotes the perturbation applied to the original orientation. The y-axis denotes the residual orientation after correction. Ideally, the final orientation ought to be the same as the original orientation, i.e. the residual pan,tilt, and roll angles would all be zero.

Examination of these plots reveal that the points appear to be displaced from the x-axis by a similar offset. This is especially the case for Scenario 10. This is due to the fact that ground truth values are not accurate. To work around this, we compute a post-correction image correlation fro each image, which indicates the extent to which the rendered image matches

the original CCD image. The image with the best correlation is selected, and the rotations of the remaining 26 cases are computed, with this orientation as the base. A summary of the post-correlation adjusted residual rotations is included in Table 3.

		Residual Rotation	
	Initial	Sc. 13	Sc. 10
Average	4.241	0.580	0.911
Minimum	0	0	0
Maximum	5.774	1.710	1.406
Median	5.385	0.536	1.114
σ	1.767	0.381	0.443

Table 3: Summary of Adjusted Residual Rotations before and after orientation correction. Angles are measured in degrees.

Based on analysis of the orientation error data, and visual examination of the matches determined by the MGA, we have arrived at the following observations.

- In the case of horizons that are reasonably distinct, such as the ones in Scenarios 13 and 10, we have obtained accurate results.
- Owing to the highly irregular nature of horizon lines, the extracted lines are highly fragmented, in spite of considerable smoothing. This renders the matching problem even more difficult.
- The constraint on minimum line length notwithstanding, a very large number of extraneous line segments are usually present, greatly increasing noise. When the horizon structure is not distinct, it is easy for the matcher to piece together a competitive false match from the extraneous line segment clusters.
- An inherent problem in horizon-line matching is that of ambiguity. Horizon lines are often self-similar in structure. Omission error increases as segments of the model corresponding to out-of-view data segments are left unmatched. This could drive the matching algorithm to find scaled-down matches in self-similar regions of the horizon. We can reduce the chances of such mismatches by discouraging scaling. Ideally, the matching problem under consideration does not involve perceptible scale change. However, there may be small variations in scale due to inaccuracies in ground truth. Also, as explained in [Beveridge, 1993], fitting line models subject to rigid transforms is actually more difficult than for variable size models.

References

[Beveridge *et al.*, 1990] J. Ross Beveridge, Rich Weiss, and Edward M. Riseman. Combinatorial Optimization Applied to Variable Scale 2D Model Matching. In *Proceedings of the IEEE International Conference on Pattern Recognition 1990, Atlantic City*, pages 18 – 23. IEEE, June 1990.

[Beveridge, 1993] J. Ross Beveridge. *Local Search Algorithms for Geometric Object Recognition: Optimal Correspondence and Pose.* PhD thesis, University of Massachusetts at Amherst, May 1993.

[Burns *et al.*, 1986] J. B. Burns, A. R. Hanson, and E. M. Riseman. Extracting straight lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI–8(4):425 – 456, July 1986.

[Whitley *et al.*, 1997] D. Whitley, J. R. Beveridge, C. Guerra-Salcedo and C. Graves. Messy Genetic Algorithms for Subset Feature Selection. In *Proc. 1997 International Conference on Genetic Algorithms*, page (submitted), July 1997.

[Goldberg *et al.*, 1989] David E. Goldberg, Bradley Korb, and Kalyanmoy Deb. Messy genetic algorithms: Motivation, analysis, and first results. Technical report, University of Alabama, May 1989.

[Horn, 1987] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of Optical Society of America*, 4:629–642, 1987.

[Beveridge *et al.*, 1996] J. Ross Beveridge and Christopher Graves and Christopher E. Lesher. Local Search as a Tool for Horizon

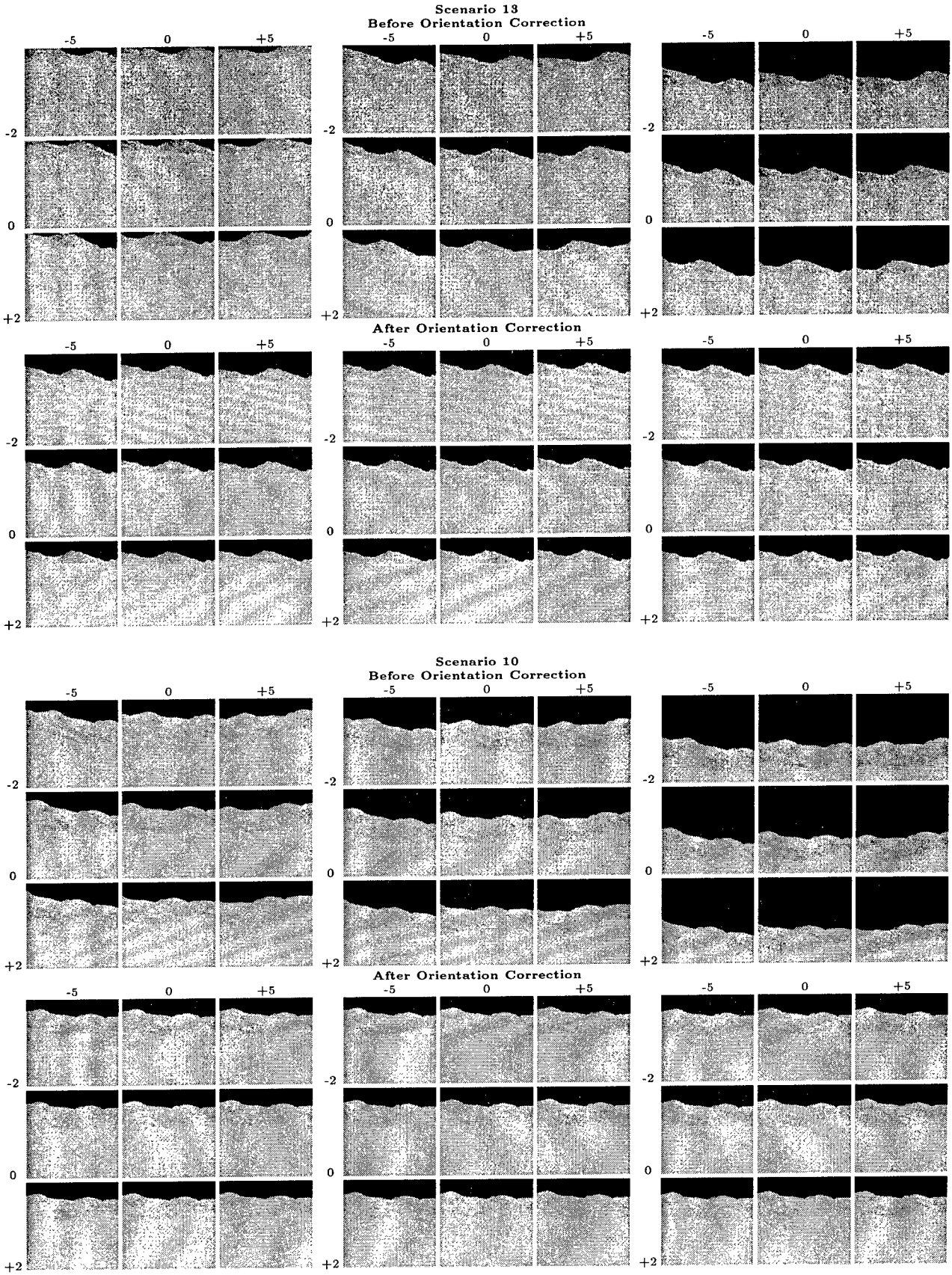


Figure 6: Terrain map renderings, before and after orientation correction x-axis: $\Delta roll$, y-axis: Δpan. $\Delta tilt$ values (left to right): $-1, 0, +2$

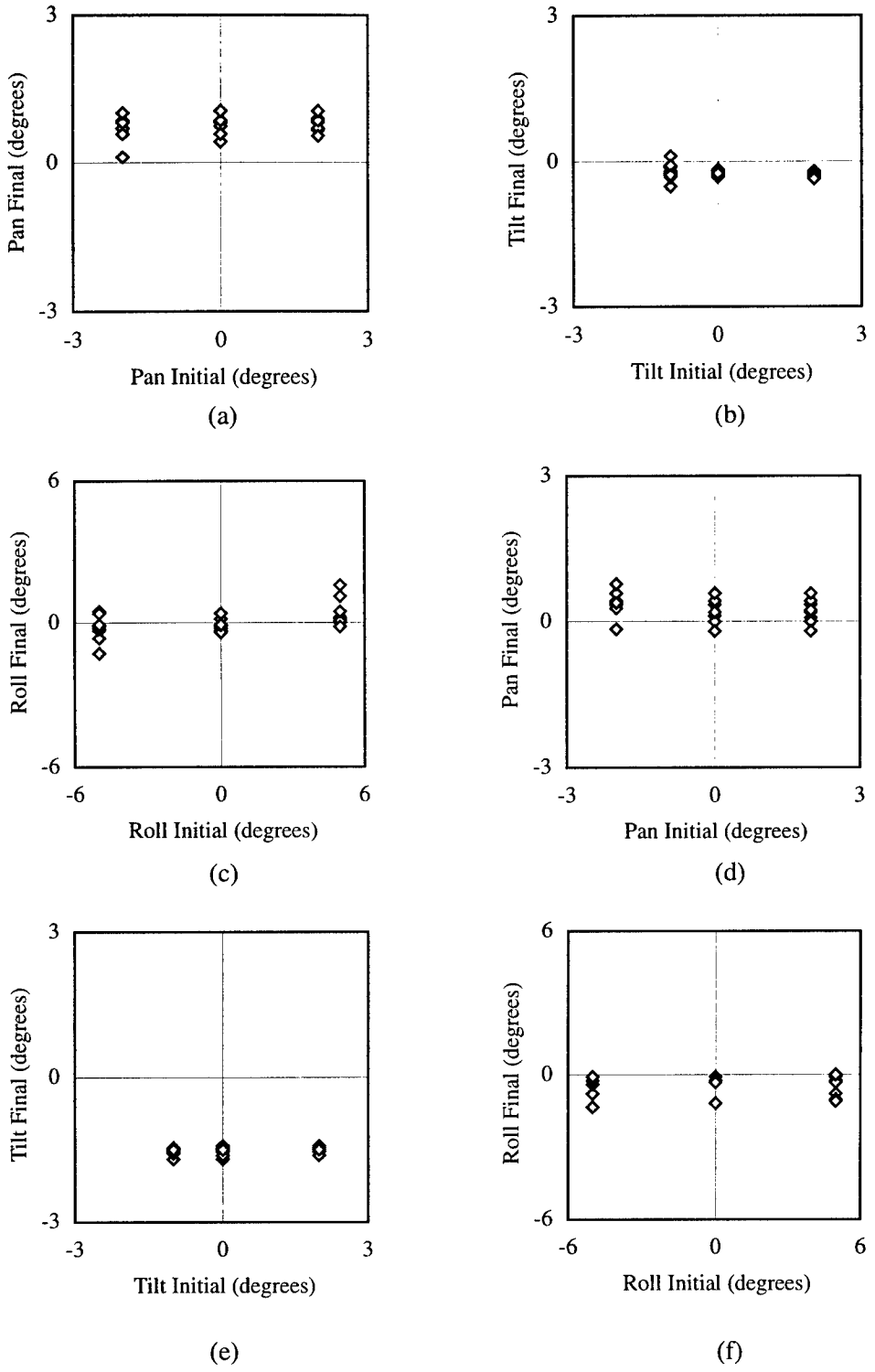


Figure 7: Residual Rotation versus Perturbation. Scenario 13: (a)-(c), Scenario 10: (d)-(f)

Line Matching. In *Proceedings: Image Understanding Workshop*, pages 683 – 686, Los Altos, CA, February 1996. ARPA, Morgan Kaufmann.

[Beveridge, 1997] J. Ross Beveridge, Edward M. Riseman and Christopher R. Graves. How Easy is Matching 2D Line Models Using Local Search? *IEEE Trans. on Pattern Analysis and Machine Intelligence*, page (to appear), 1997.

[Rimey and Hougen, 1995] Discussion of SSV Orientation Correction with Lockheed-Martin RSTA Group. Personal Correspondence, 1995.

[Rimey, 1995] RSTA Sept94 Data Collection Final Report. Technical report, Martin Marietta Astronautics, Denver, CO, January 1995.

[Sutherland and Thompson, 1994] Sutherland, K.T. and Thompson, W.B. Localizing in Unstructured Environments: Dealing with the Errors. *Robotics and Automation*, 10:740–754, 1994.

[Thompson *et al.*, 1993] W.B. Thompson, T.C. Henderson, T.L. Colvin, L.B. Dick, and C.M. Valiquette. Vision-Based Localization. In *Proceedings: Image Understanding Workshop*, pages 491–498, Los Altos, CA, 1993. ARPA, Morgan Kaufmann.

Fast Image Stabilization and Mosaicking

Carlos Morimoto **Rama Chellappa** **Stephen Balakirsky**
Center for Automation Research U.S. Army Research Laboratory
University of Maryland Adelphi, MD 20783
College Park, MD 20742

Abstract

We present two fast implementations of electronic image stabilization and mosaicking systems. The first one is based on a 2D similarity model and is targeted to process PREDATOR video data. The second system uses a 3D model and compensates for 3D rotation. Both systems have been implemented on parallel pipeline image-processing hardware (a Datacube MaxVideo 200) connected to a Themis 10MP. Both algorithms use a feature-based multiresolution technique which tracks a small set of features to estimate the motion of the camera. The extended Kalman filter framework is employed by the 3D derotation system. The inter-frame motion estimates relative to a reference frame are used to warp the current frame in order to achieve stabilization. The estimates are also used to construct mosaics by aligning the frames. A fast mosaicking implementation is presented for the 2D system. Experimental results demonstrate the robustness of both systems at frame rates above 10 frames/second.

1 Introduction

Camera motion estimation is an integral part of any computer vision or robotic system that has to navigate in a dynamic environment. Whenever part of the camera motion is not necessary or "unwanted", image stabilization can be applied as a preprocessing step before further analysis of the image sequence. It can be used as a front-end system in a variety of dynamic image analysis applications or simply as a visualization tool. Image stabilization has been used for the computation of egomotion [Viéville *et al.*, 1993; Irani *et al.*, 1994], video compression [Kwon *et al.*, 1995; Morimoto *et al.*, 1996], and detection and tracking of independently moving objects [Balakirsky, 1995; Burt and Anandan, 1994; Morimoto *et al.*, 1995]. For more natural visualization, vehicle models are used to filter high-frequency or oscillatory motion components due to irregularities of the terrain [Durić and Rosenfeld, 1995; Yao *et al.*, 1995].

Methods proposed for electronic image stabilization can be distinguished by the models adopted to estimate the camera motion. Several 2D and 3D stabilization schemes are described by Davis *et al.* [Davis *et al.*, 1994]. For 2D models, all the estimated motion parameters are in general compensated for, i.e., all motion is removed from the input sequence [Burt and Anandan, 1994; Irani *et al.*, 1995; Sawhney *et al.*, 1995].

For 3D models under perspective projection, the displacement of each image pixel will also depend on the structure of the scene, or more precisely, on the depth of the corresponding 3D point. It is possible to parameterize these models so that only the translational components carry structural information, while the rotational components are depth-independent. Stabilization in 3D is achieved by derotating the frames, generating a translation-only sequence, or a sequence containing only translation and low-frequency rotation (smoothed rotation) [Durić and Rosenfeld, 1995; Yao *et al.*, 1995]. By compensating for the camera rotation, the resulting image sequence looks mechanically stabilized, as if the camera were mounted on a gyroscopic platform.

Most of the current image stabilization algorithms that have been implemented in real time use 2D models due to their simplicity [Hansen *et al.*, 1994; Morimoto and Chellappa, 1996]. Feature-based motion estimation or image registration algorithms are used by these methods in order to bring all the images in the sequence into alignment. These algo-

The support of the Defense Advanced Research Projects Agency (ARPA Order No. A422) and the U.S. Army Research Office under Contract DAAH04-93-G-0419 is gratefully acknowledged.

rithms are targeted to specific real-time image processing platforms. The systems operate with images of resolution 128×128 at above 10 frames per second, and are robust to large image displacements. The system developed by Hansen *et al.* [Hansen *et al.*, 1994] uses a mosaic-based registration technique implemented on pyramidal hardware (VFE-100). The system uses a multi-resolution, iterative process to estimate the affine motion parameters between levels of Laplacian pyramid images. From coarse to fine levels, the optical flow of local patches of the image is computed using a cross-correlation scheme. The motion parameters are then computed by fitting an affine motion model to the flow. These parameters are used to warp the previous image of the next finer pyramid level to the current image, and the refinement process continues until the desired precision is achieved. This scheme, combined with the construction of a mosaic image, allows the system to cope with large image displacements.

In this paper we present two image stabilization systems based on 2D and 3D models. The 2D system includes several modifications to the system presented in [Morimoto *et al.*, 1995], in order to process data from PREDATOR video, which are sequences taken from an airborne platform and are characterized by low quality and relatively low inter-frame displacement (less than 10% of the image size). Most of the modifications were necessary because of the low quality of the video sequences, due to lossy compression. The second system uses an extended Kalman filter (EKF) to estimate the 3D motion of the camera, and stabilization is achieved by derotating the input sequence.

Both systems were implemented on a Datacube MaxVideo 200 card plugged into the same VME backplane as a Themis 10MP. The MV200 is a parallel pipeline image processing board very commonly used for real-time image processing, and the Themis is a dual 100MHz hyperSPARC board which is running Solaris 2.4. They are able to process about 10 frames per second for 8-bit gray level images of resolution 128×120. The 2D system is also able to construct mosaic images in real time, directly onto a window on the host computer.

This paper is organized as follows. Section 2 introduces the 2D model-based image stabilization algorithm; the 3D algorithm is described in Section 3. Section 4 describes the implementation of the real-time 2D mosaicking display and how 3D mosaics can also be computed. Section 5 shows experimental results of the performance of both systems, and Section 6 concludes the paper.

2 2D Image Stabilization Algorithm

The 2D similarity model-based image stabilization system is based on the fast implementation of the image stabilization algorithm presented in [Morimoto *et al.*, 1995]. A basic stabilization system is composed of three modules shown in Figure 1. The *motion estimation* module computes the motion or global transformation between consecutive frames which is used by the *motion compensation* module to determine the global transformation which brings the newest frame into alignment with the reference frame. The *image composition* module generates the stabilized sequence and/or mosaic by warping the current frame using the motion estimates. Section 4 describes how the motion estimates are also used to construct a mosaic in real time by directly aligning the current frame with the mosaic constructed from previous frames.

Figure 1: Modules of a general electronic image stabilization system.

A block diagram of the 2D system is shown in Figure 2. The modules inside the dotted line are performed by the Datacube board, while the other modules are processed by the host computer. The Datacube digitizes the video signal from the camera and builds Gaussian and Laplacian pyramids for each new frame. The Laplacian pyramid is used for feature detection and tracking is performed on the Gaussian images. Feature detection and tracking, motion estimation, motion compensation, and mosaic construction are done by the host computer. The Datacube also receives the computed global motion to warp the current frame and generates the stabilized video output.

2.1 Motion Estimation

The structure of the motion estimation module is similar to the feature-based multi-resolution image registration algorithm presented in [Zheng and Chellappa, 1993]. Starting from the coarsest Laplacian pyramid level, a small number of non-overlapping regions are scanned, and the pixel with maximum intensity in each region is selected for tracking. Each feature is tracked between the corresponding Gaussian pyramid level of the current and previous frames, using the sum of absolute differences (SAD) as similarity measure.

The SAD between two windows of size $2W + 1$ centered at feature point $P_t(x, y)$ and its matching candidate $P_{t-1}(u, v)$ is given by

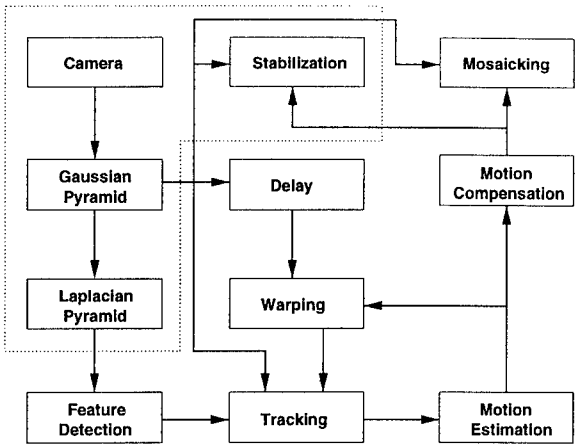


Figure 2: Block diagram of the 2D stabilization and mosaicking system.

$$SAD = \sum_{ij} |(P_t(x+i, y+j) - P_{t-1}(u+i, v+j)| \quad (1)$$

where i and j vary from $-W$ to $+W$. A match is obtained by searching for the minimum SAD over a neighborhood (search window) around the feature. For a feature at pixel coordinates (x, y), the search is performed by varying the candidate coordinates (u, v) in the interval $[(x-S)\ldots(x+S), (y-S)\ldots(y+S)]$, where $2S+1$ defines the search window size. After the grid-to-grid matches are obtained, displacements with subpixel accuracies are computed using a differential method [Tian and Huhns, 1986]. Subpixel accuracy is necessary to eliminate the quantization error introduced when the images are digitized.

The feature displacements are then used to fit a four-parameter similarity model defined by

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \mathcal{S} \begin{pmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{pmatrix} \begin{pmatrix} x_{i-1} \\ y_{i-1} \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \quad (2)$$

where (x_i, y_i) are the image frame coordinates at time t_i, $(\Delta x \ \ \Delta y)^t$ is the translation vector measured in the image coordinate system of the frame at t_i (f_i), Θ is the rotation angle between the two frames and $\mathcal{S}$ is the scaling factor. Notice that $\mathcal{S}$ is inversely proportional to the ratio of the distances between two arbitrary image points at times t_i, t_{i-1}. Thus $\mathcal{S}$ can be computed given a set of matched points from both frames, independently of the translation and rotation between them.

The scaling factor $\mathcal{S}$ is estimated first by computing the ratio of the distances in the feature sets relative to their center of mass. Assuming small rotation, the trigonometric terms in (2) can be linearized to compute the remaining translation and rotation parameters. A system of linear equations is then obtained by substituting all N matched feature pairs into the linearized equations. Each pair introduces two equations; hence the linear system has $2N$ equations and three unknowns (Θ, ΔX, and ΔY), and can be solved by a least-squares method.

The motion parameters obtained from the coarsest pyramid level are used to warp the next higher pyramid level of the previous Gaussian pyramid, and the process of tracking, estimation, and warping repeats until the highest-resolution image is reached. For an arbitrary pyramid level, the new estimate must be combined with the previous coarser-level estimate before warping the image at the next higher pyramid level (an initial zero motion is assumed for the coarsest level). Assuming that the total motion estimate from the coarser levels is $M_{i-1} = (\Delta x_{i-1}, \Delta y_{i-1}, \Theta_{i-1}, \mathcal{S}_{i-1})$ and the estimate for the current level is $m_i = (d_x, d_y, \theta, s)$, the new total motion estimate M_i used to warp the next-higher-resolution image can be easily derived to be [Zheng and Chellappa, 1993]

$$M_i = (\Delta x_i \Delta y_i \Theta_i \mathcal{S}_i)^T = \begin{pmatrix} s\cos\theta\Delta x_{i-1} - s\sin\theta\Delta y_{i-1} + d_x \\ s\sin\theta\Delta x_{i-1} + s\cos\theta\Delta y_{i-1} + d_y \\ \theta + \Theta_{i-1} \\ s \times \mathcal{S}_{i-1} \end{pmatrix} \quad (3)$$

2.2 Motion Compensation

The motion compensation module keeps a history of the inter-frame motion to remove what is unwanted and compute the warping parameters that will stabilize the current image frame. One of the advantages of electronic image stabilization systems is that motion can be compensated on demand, offering great flexibility by simply modifying some parameters of the compensation module.

The motion compensation module keeps track of the total combined motion, from the reference frame up to the current frame. When a new estimate is sent from the motion estimation module, the total motion is updated using (3).

Our system does not perform temporal smoothing on any of the motion parameters, but allows the user to dynamically mask (enable/disable) each parameter independently, for display purposes. For example, when the camera moves forward, producing a divergent image flow, the computed transformation includes a reduction in scale, which basically eliminates the perception of forward motion by producing a shrinking image with internal zero flow. The forward motion perception is restored by simply masking the scaling factor on display.

3 3D Image Stabilization Algorithm

The 3D model-based stabilization algorithm uses an extended Kalman filter (EKF) to estimate the rotation between frames, which is represented using unit quaternions. A small set of feature points is tracked as described previously, except that no estimation refinement is computed between pyramid levels, i.e., the features are simply scaled between levels, in a similar way to the greedy multi-resolution search implemented in [Morimoto and Chellappa, 1996].

3.1 Camera Motion Model

Let $\mathbf{P} = (X, Y, Z)^T$ be a 3D point in a Cartesian coordinate system fixed on the camera and let $\mathbf{p} = (x, y)^T$ be the corresponding image position of $\mathbf{P}$ (see Figure 3). The image plane defined by the x and y axes is perpendicular to the Z axis and contains the principal point $(0, 0, f)$. Thus the perspective projection of a point $\mathbf{P} = (X, Y, Z)^T$ onto the image plane is

$$\mathbf{p} = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f\frac{X}{Z} \\ f\frac{Y}{Z} \end{pmatrix} \quad (4)$$

where f is the camera focal length.

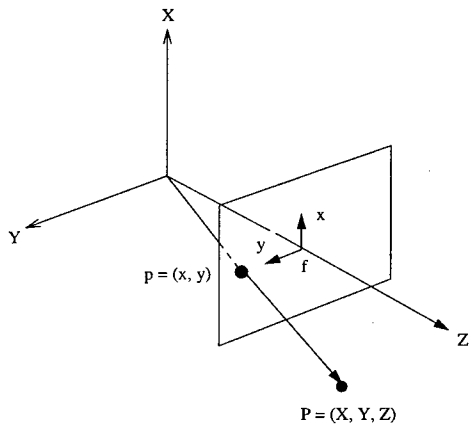


Figure 3: Diagram of the coordinate system fixed on the camera.

Under a rigid motion assumption, an arbitrary motion of the camera can be described by a translation $\mathbf{T}$ and a rotation $\mathbf{R}$, so that a point $\mathbf{P}_0$ in the camera coordinate system at time t_0 has a new camera position at time t_1 given by

$$\mathbf{P}_1 = \mathbf{R}\mathbf{P}_0 + \mathbf{T} \Rightarrow$$
$$\begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix} + \begin{pmatrix} T_X \\ T_Y \\ T_Z \end{pmatrix} \quad (5)$$

where $\mathbf{R}$ is a 3×3 orthonormal matrix.

The projection of $\mathbf{P}_1$ can be obtained by combining (4) and (5):

$$\mathbf{p}_1 = \begin{pmatrix} f\frac{X_1}{Z_1} \\ f\frac{Y_1}{Z_1} \end{pmatrix} = \begin{pmatrix} f\frac{(r_{11}x_0 + r_{12}y_0 + r_{13}f + f\frac{T_X}{Z_0})}{(r_{31}x_0 + r_{32}y_0 + r_{33}f + f\frac{T_Z}{Z_0})} \\ f\frac{(r_{21}x_0 + r_{22}y_0 + r_{23}f + f\frac{T_Y}{Z_0})}{(r_{31}x_0 + r_{32}y_0 + r_{33}f + f\frac{T_Z}{Z_0})} \end{pmatrix} \quad (6)$$

For distant points ($Z_0 \gg T_X, T_Y$, and T_Z), the displacement is basically due to rotation, and thus (6) simplifies to

$$\mathbf{p}_1 = \begin{pmatrix} f\frac{(r_{11}x_0 + r_{12}y_0 + r_{13}f)}{(r_{31}x_0 + r_{32}y_0 + r_{33}f)} \\ f\frac{(r_{21}x_0 + r_{22}y_0 + r_{23}f)}{(r_{31}x_0 + r_{32}y_0 + r_{33}f)} \end{pmatrix} \quad (7)$$

The use of distant features for motion estimation and image stabilization has been addressed before in [Davis *et al.*, 1994; Durić and Rosenfeld, 1995; Yao *et al.*, 1995]. Such features constitute very strong visual cues that are present in almost all outdoor scenes, although it might be hard to guarantee that all the features are distant. In this paper we estimate the three parameters that describe the rotation of the camera using an iterated extended Kalman filter (IEKF).

3.1.1 Quaternions

Common ways to represent rotation include 3×3 orthonormal matrices, Euler angles, axis plus angle, and unit quaternions [Kanatani, 1990]. Young and Chellappa [Young and Chellappa, 1990] used quaternions for the problem of 3D motion estimation from noisy stereo sequences, and Horn [Horn, 1987] used quaternions to solve the absolute orientation problem from three or more point correspondences. Many other applications of quaternions can be found in photogrammetry, robotics and computer vision because of their compactness and good numerical properties, which facilitate the process of rotation estimation.

Quaternions are 4-tuples (q_0, q_x, q_y, q_z) that can be interpreted as complex numbers with one real (q_0) and three imaginary parts (q_x, q_y, q_z), as a scalar plus a 3D vector, or simply as a vector in 4D-space. To see how quaternions can represent rotations, consider a 3D unit sphere defined by $X^2 + Y^2 + Z^2 = 1$. The position of a point on the surface of the sphere can directly represent pan and tilt but not roll. By introducing a fourth parameter, it is now possible to represent an arbitrary 3D rotation by a point on a 4D unit sphere where $q_0^2 + q_x^2 + q_y^2 + q_z^2 = 1$.

3.1.2 Relevant Properties of Unit Quaternions

In this section we present only a few basic properties of quaternions. More detailed treatments can be found in [Horn, 1987; Kanatani, 1990]. Consider

a quaternion as composed of a scalar and a vector part, as in

$$\breve{\mathbf{q}} = q_0 + \mathbf{q}, \qquad \mathbf{q} = (q_x, q_y, q_z)^T \tag{8}$$

The *dot product* operator for quaternions is defined as $\breve{\mathbf{p}}.\breve{\mathbf{q}} = p_0 q_0 + \mathbf{p}.\mathbf{q}$, and the *norm* of a quaternion is given by $|\breve{\mathbf{q}}| = \breve{\mathbf{q}}.\breve{\mathbf{q}} = q_0^2 + \mathbf{q}.\mathbf{q}$. Unit quaternions are simply defined as quaternions with unit norm.

A unit quaternion can be interpreted as a rotation θ around a unit vector $\mathbf{w}$ using the equation $\breve{\mathbf{q}} = \sin(\theta/2) + \mathbf{w}\cos(\theta/2)$. Note that $\breve{\mathbf{q}}$ and $-\breve{\mathbf{q}}$ correspond to the same rotation since a rotation of θ around a vector $\mathbf{q}$ is equivalent to a rotation of $-\theta$ around the vector $-\mathbf{q}$.

Let the conjugate of a quaternion be defined as

$$\breve{\mathbf{q}}^* = q_0 - \mathbf{q} \tag{9}$$

and the multiplication of two quaternions as

$$\breve{\mathbf{r}} = \breve{\mathbf{p}}\breve{\mathbf{q}} = \begin{cases} r_0 = p_0 q_0 - \mathbf{p}.\mathbf{q}; \\ \mathbf{r} = p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q}. \end{cases} \tag{10}$$

Thus, the conjugate of $\breve{\mathbf{q}}$ is also its inverse since $\breve{\mathbf{q}}^*\breve{\mathbf{q}} = \breve{\mathbf{q}}\breve{\mathbf{q}}^* = 1$. It is useful to have the multiplication of quaternions expanded in matrix form as follows:

$$\breve{\mathbf{p}}\breve{\mathbf{q}} = \begin{pmatrix} p_0 & -p_x & -p_y & -p_z \\ p_x & p_0 & -p_z & p_y \\ p_y & p_z & p_0 & -p_x \\ p_z & -p_y & p_x & p_0 \end{pmatrix} \begin{pmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{pmatrix} \tag{11}$$

Note that the multiplication of quaternions is associative but it is not commutative, i.e., in general $\breve{\mathbf{p}}\breve{\mathbf{q}}$ is not the same as $\breve{\mathbf{q}}\breve{\mathbf{p}}$.

The rotation of a vector or point $\mathbf{P}$ to a vector or point $\mathbf{P}'$ can be represented by a quaternion $\breve{\mathbf{q}}$ according to

$$(0 + \mathbf{P}') = \breve{\mathbf{q}}(0 + \mathbf{P})\breve{\mathbf{q}}^* \tag{12}$$

Composition of rotations can be performed by multiplication of quaternions since

$$\begin{gathered}(0 + \mathbf{P}'') = \breve{\mathbf{q}}(0 + \mathbf{P}')\breve{\mathbf{q}}^* = \\ \breve{\mathbf{q}}(\breve{\mathbf{r}}(0 + \mathbf{P})\breve{\mathbf{r}}^*)\breve{\mathbf{q}}^* = (\breve{\mathbf{q}}\breve{\mathbf{r}})(0 + \mathbf{P})(\breve{\mathbf{q}}\breve{\mathbf{r}})^*\end{gathered} \tag{13}$$

where it is easy to verify that $(\breve{\mathbf{r}}^*\breve{\mathbf{q}}^*)$ is equivalent to $(\breve{\mathbf{q}}\breve{\mathbf{r}})^*$.

The nine components of the orthonormal rotation matrix $\mathbf{R}$ in (5) can be represented by the parameters of a unit quaternion simply by expanding (12) using (11), so that

$$\mathbf{R} = \begin{pmatrix} 1-2q_y^2-2q_z^2 & 2(-q_0q_z+q_xq_y) & 2(q_0q_y+q_xq_z) \\ 2(q_0q_z+q_xq_y) & 1-2q_x^2-2q_z^2 & 2(-q_0q_x+q_yq_z) \\ 2(-q_0q_y+q_xq_z) & 2(q_0q_x+q_yq_z) & 1-2q_x^2-2q_y^2 \end{pmatrix} \tag{14}$$

3.2 3D Motion Estimation

The dynamics of the camera is described as the evolution of a unit quaternion, and an IEKF is used to estimate the inter-frame rotation $\breve{\mathbf{q}}$. EKFs have been extensively used for motion estimation from a sequence of images [Broida and Chellappa, 1986; Yao *et al.*, 1995; Young and Chellappa, 1990]. In order to achieve real-time performance, this framework was simplified to compute only the rotational parameters from distant feature points.

A unit quaternion has only three degrees of freedom due to its unit norm constraint, so that it will be represented using only the vector parameters. The remaining scalar parameter is computed from

$$q_0 = (1 - q_x^2 - q_y^2 - q_z^2)^{\frac{1}{2}} \tag{15}$$

Only nonnegative values of q_0 in (15) are considered, so that (14) can be rewritten using (q_x, q_y, q_z) only.

The state vector $\mathbf{x}$ and plant equations are defined as follows:

$$\left.\begin{array}{c} \mathbf{x} \stackrel{def}{=} \breve{\mathbf{q}} + \mathbf{n} \\ \dot{\mathbf{x}} = \mathbf{0} \end{array}\right\} \Rightarrow \mathbf{x}(t_{i+1}) = \mathbf{x}(t_i) \tag{16}$$

The following measurement equations are derived from (7):

$$\mathbf{z}(t_i) = \mathbf{h}_{i|i-1}[\mathbf{x}(t_i)] + \eta(t_i) \tag{17}$$

where $\mathbf{h}$ is a nonlinear function which relates the current state to the measurement vector $\mathbf{z}(t_i)$, and η is the measurement noise. After tracking a set of N feature points, a two-step EKF algorithm is used to estimate the total rotation. The first step is to compute the state and covariance predictions at time t_{i-1} before incorporating the information from $\mathbf{z}(t_i)$ by

$$\begin{gathered}\hat{\mathbf{x}}(t_i^-) = \hat{\mathbf{x}}(t_{i-1}^+) \\ \boldsymbol{\Sigma}(t_i^-) = \boldsymbol{\Sigma}(t_{i-1}^+) + \boldsymbol{\Sigma}_{\mathbf{n}}(t_{i-1})\end{gathered} \tag{18}$$

where $\hat{\mathbf{x}}(t_{i-1}^+)$ is the estimate of $\mathbf{x}(t_{i-1})$ and $\boldsymbol{\Sigma}(t_{i-1}^+)$ is its associated covariance obtained based on the information up to time t_{i-1}; $\hat{\mathbf{x}}(t_i^-)$ and $\boldsymbol{\Sigma}(t_i^-)$ are the predicted estimates before the incorporation of the i^{th} measurements; and $\boldsymbol{\Sigma}_{\mathbf{n}}(t_{i-1})$ is the covariance of the plant noise $\mathbf{n}(t_{i-1})$.

The *update step* follows the previous *prediction step*. When $\mathbf{z}(t_i)$ becomes available, the state and covariance estimates are updated by

$$\begin{gathered}\mathbf{K}(t_i) = \frac{\boldsymbol{\Sigma}(t_i^-)\mathbf{H}_{i|i-1}^T}{\mathbf{H}_{i|i-1}\boldsymbol{\Sigma}(t_i^-)\mathbf{H}_{i|i-1}^T + \boldsymbol{\Sigma}_\eta(t_i)} \\ \hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)\{\mathbf{z}(t_i) - \mathbf{h}_{i|i-1}[\hat{\mathbf{x}}(t_i^-)]\} \\ \boldsymbol{\Sigma}(t_i^+) = [\mathbf{I} - \mathbf{K}(t_i)\mathbf{H}_{i|i-1}]\boldsymbol{\Sigma}(t_i^-)\end{gathered} \tag{19}$$

where $\mathbf{K}(t_i)$ is a $3 \times N$ matrix which corresponds to the Kalman gain, $\mathbf{\Sigma}_\eta(t_i)$ is the covariance of $\eta(t_i)$ and $\mathbf{I}$ is a 3×3 identity matrix. $\mathbf{H}_{i|i-1}$ is the linearized approximation of $\mathbf{h}_{i|i-1}$, defined as

$$\mathbf{H}_{i|i-1} = \left.\frac{\delta \mathbf{h}_{i|i-1}}{\delta \mathbf{x}(i)}\right|_{\hat{\mathbf{x}}(t_i^-)} \tag{20}$$

A batch process using a least-square estimate of the rotational parameters can be used to initialize the algorithm, as in [Yao *et al.*, 1995], but for the experiments shown in Section 5 we simply assume zero rotation as the initial estimate.

To speed up the process and reduce the amount of computation that is required to achieve real-time performance, the measurement vectors are sorted according to their fits to the previous estimate, so that only the best M points ($M < N$) are actually used by the EKF. The solution is refined iteratively by using the new estimate $\hat{\mathbf{x}}_i$ to evaluate $\mathbf{H}$ and $\mathbf{h}$, for a few iterations. More detailed derivations of the Kalman filter equations can be found in [Jazwinski, 1970; Maybeck, 1982].

4 Fast 2D Mosaicking Implementation

The goal of creating a mosaic image is to compose a picture which has a larger field of view than the input sensor. This may be accomplished by combining individual frames which depict portions of the overall scene. By appropriately pasting these frames together, we can generate a panoramic view of the original scene. We refer to this procedure as mosaicking.

Through the use of the motion stabilization technique described in Section 2, it is possible to not only compensate for inter-frame motion, but also to keep track of the motion parameters which represent image frame changes with respect to a global coordinate system. We are then able to paste each incoming frame into our global picture. This process generates a mosaic of the imaged scene.

A fast mosaicking system for PREDATOR video data was implemented as an extension to the stabilization system, and runs on the same platform. This extended system can be broken down into two distinct processing threads: a real-time thread for performing the image processing in conjunction with the Datacube card, and a mosaic thread which controls the display and user interface. The two threads communicate with each other through UNIX pipes. By using an X-Windows interface, the actual display head for the mosaic may be thousands of miles from the processor. This may be accomplished by setting the display head to be any X-Windows-compatible terminal which is networked to the mosaic processor.

The last stage of the real-time thread warps the highest pyramid level of the current frame to be an integer offset from the global coordinate frame. This warp is accomplished on the Datacube through the use of an affine transform, and removes all scale, rotation, and fractional pixel shifts. This warped image is then sent to the display process along with its location in the global system. The display process starts with a blank global image, and adds each incoming image segment to this image.

In our current implementation of the mosaicking process, the global image is treated as a write-once medium which is unlimited in size. There are two reasons behind this. The first is that the error in the stabilization algorithm is cumulative. Therefore, if the camera dwells on a particular area for an extended period of time, image drift which distorts the mosaic is noticeable. By providing a write-once memory, mosaic discontinuities are confined to a single line in the image. This makes for a much more viewable mosaic. The second reason for the write-once memory is to reduce the display channel bandwidth. It is possible to foresee applications where the image processor may be located at a remote ground (or air) station and may only be connected to the user's console by a low-bandwidth link. We hope to show that our mosaic creation technique allows for realistic scene generation while consuming little communication bandwidth.

In order to avoid running out of room for the generation of a mosaic, the global image is treated as if it were unlimited in size. In reality, the image storage size is constrained by the user, and the image is scrolled to maintain the current view on the screen. Areas which scroll off the edge of the display are lost.

5 Experimental Results

Since still images are not the most appropriate way of displaying the results of a dynamic process such as stabilization, we have made the original, stabilized and mosaicked sequences available in MPEG format at **http://www.cfar.umd.edu/~carlos/IUW97.html**. In the following sections, an MPEG file at this address named *Mosaic* will be referenced by http:Mosaic.

5.1 2D Stabilization and Mosaicking Results

This section presents experimental results from applying the fast 2D stabilization and mosaicking system to PREDATOR video data. Our data tape has been through several recording generations, and is of moderate to poor quality.

Figure 5: Mosaic from the 2D fast stabilization algorithm

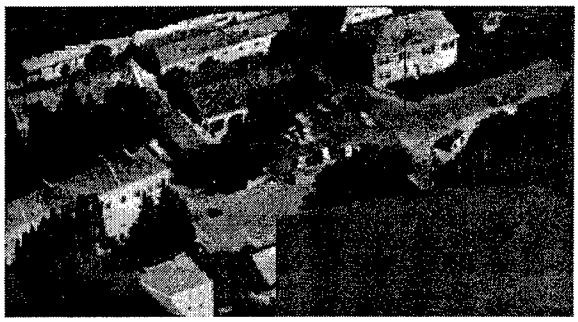


Figure 4: Mosaic from the 2D fast stabilization algorithm

Figures 4 and 5 show mosaic images from PREDATOR video data using the mosaicking process described above. Reliable image stabilization requires large overlap between image frames and reasonably high frame rates. Fortunately this is significantly less important for the display thread. Since the mosaic is treated as write-once memory, overlapping areas of the image are ignored. Therefore, it is necessary that there be only a small overlap between frames in order to provide a continuous mosaic. This allows us to discard frames from the real-time thread without displaying them. Running on the hardware described above, our real-time thread was able to run at approximately 10 frames per second. The display thread was set up to process every fourth image which was generated by the real-time thread. The rest of the images were discarded.

The images used to generate the mosaic were from the top level of the image pyramid. This corresponds to an image resolution of 128x120 pixels. The bandwidth which would be necessary to transmit the mosaic in real time is image sequence dependent. The use of the write-once memory dictates that the required bandwidth is directly related to the amount of new information contained in each mosaic frame. For the mosaics shown in Figures 4 and 5, we recorded an average bandwidth of 15993 bytes per second, for a two-frame-per-second mosaic update rate. This is an 89% improvement over sending the entire raw sequence.

5.2 3D Stabilization and Mosaicking Results

Figure 6 is an example of the results obtained from our real-time 3D derotation system using an off-road sequence provided by NIST. The camera is rigidly mounted on the vehicle and is moving forward. The top row shows the fifth frame from the input sequence (left) and its corresponding stabilized frame (right). The original and stabilized sequences are available at http:OffRoad3DStabilization. The difference between the fifth and tenth input frames is shown on the bottom left, and the difference between the corresponding stabilized frames on the bottom

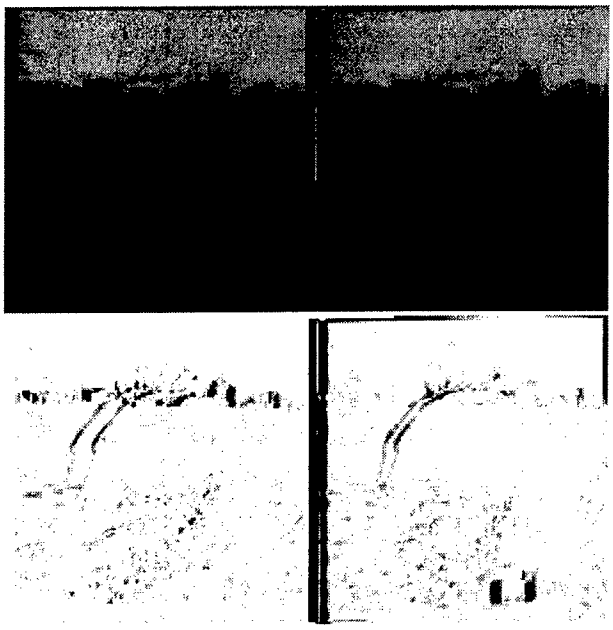


Figure 6: Image stabilization results. The top row contains the fifth frame (left) and its corresponding stabilized frame (right). The bottom row shows the difference between two frames of the input sequence (left) and the difference of the corresponding stabilized frames (right). Stabilization minimizes the difference in regions close to the horizon.

right. The darkness of a spot on the bottom images is proportional to the difference of intensity between the corresponding spots in each frame. Since stabilization has to compensate for the motion between frames, the difference images can be considered as error measurements which stabilization attempts to minimize. In this example, most of the features lie on the horizon, so that the horizon is particularly well stabilized. Errors are bigger around objects that are closer to the camera (darker regions), since they have large translational components which are not compensated by this method.

The real-time implementation typically detects and tracks nine features with a maximum feature displacement of 15 pixels. Under these settings, the system is able to process approximately 9.8 frames per second.

Figure 7 compares the mosaic images from the 2D and 3D models. They both use the same set of 11 feature points for motion estimation, and the 3D system selects the four points which best approximate the current rotation estimate to update its state.

The original sequence is composed of 200 frames and the dominant motion is left-to-right panning. To help in comparison and visualization, the reference frame was assumed to be the 100th frame, and appears at the center of the mosaics. The first column shows the 50th, 100th, and 200th frames from top to bottom. The second column shows the corresponding mosaic images constructed from the 2D estimates, and the third column shows the corresponding mosaics constructed using the 3D estimates. Since the camera calibration is unknown, we "guessed" the camera FOV to be 3×4 degrees. The mosaic from the 2D estimates (http:2DMosaic1) does a good job locally, but the 3D mosaic (http:3DMosaic1) looks much more natural, as if it were a panoramic picture taken using a fish-eye lens. The original sequence and the 3D stabilized output can be seen at http:3DStabilization1.

Figure 8 shows a second comparison between 2D and 3D mosaics. The original sequence is composed of 150 frames and the dominant motion is right-to-left panning. The reference frame was assumed to be the 75th frame, and appears at the center of the mosaics. The top row shows the 70th and the last frame of the sequence, from left to right. The second row shows the 2D mosaics constructed up to the corresponding frames in the first row, and the bottom row shows the corresponding mosaics using the 3D models and using the same camera parameters that were used to generate Figure 7. The original sequence and the mosaics can be viewed at http:UGVsequence, http:2DUGVMosaic, and http:3DUGVMosaic.

6 Conclusion

We have presented in this paper a fast electronic image stabilization and mosaicking system based on a two-dimensional feature-based multi-resolution motion estimation algorithm, that tracks a small set of features to estimate the motion of the camera. Stabilization is achieved by combining all motion from a reference frame and subtracting this motion from the current frame. Mosaics are constructed in real time by directly aligning new frames with the current mosaic. The system was implemented on a Datacube MaxVideo 200 board connected to a Themis 10MP. Preliminary tests using PREDATOR video data demonstrate the robustness of the system, which is able to process 10 frames per second and handle displacements of up to ± 12 pixels between consecutive frames.

We have also presented a 3D model-based real-time stabilization system that estimates the motion of the camera using an IEKF. Stabilization is achieved by derotating the input camera sequence. Rotations are represented using unit quaternions, whose good numerical properties contribute to the overall performance of the system. The system was implemented on the same platform and it is able to process 128×120 images at approximately 10 Hz.

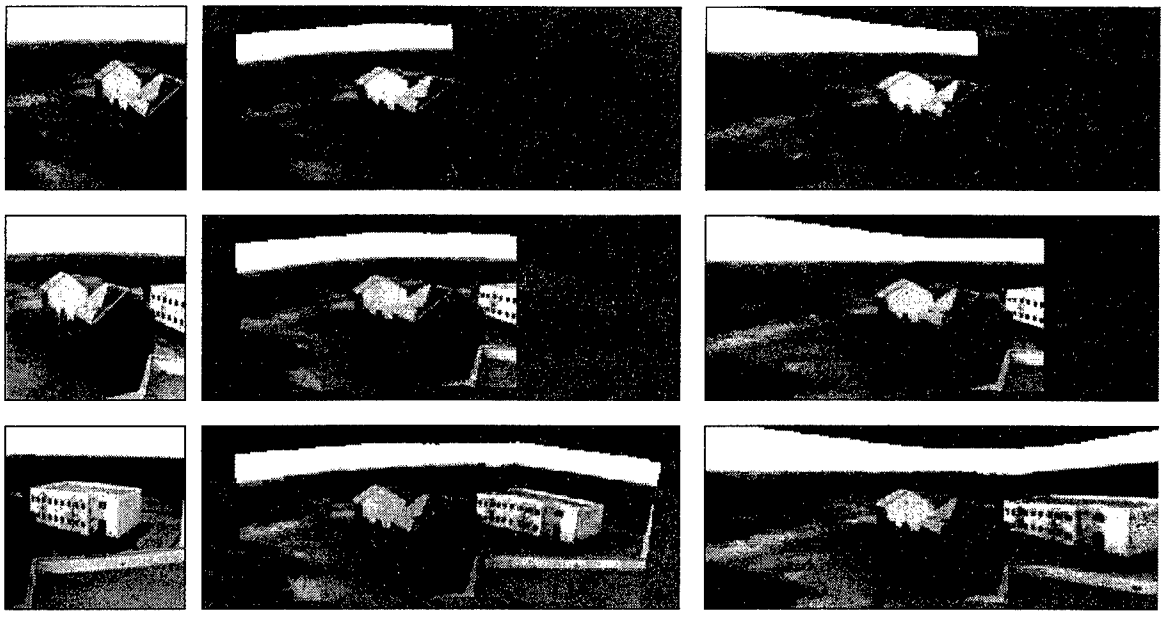


Figure 7: 2D and 3D mosaics from 200 frames of a panning sequence. The leftmost column shows the 50th, 100th, and 200th frames from the input sequence. The second and third columns show the corresponding 2D and 3D mosaics,

References

[Balakirsky, 1995] S. Balakirsky. Comparison of electronic image stabilization systems. Master's thesis, Department of Electrical Engineering, University of Maryland, College Park, 1995.

[Broida and Chellappa, 1986] T. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:90–99, 1986.

[Burt and Anandan, 1994] P. Burt and P. Anandan. Image stabilization by registration to a reference mosaic. In *Proc. DARPA Image Understanding Workshop*, pages 425–434, Monterey, CA, November 1994.

[Davis *et al.*, 1994] L.S. Davis, R. Bajcsy, R. Nelson, and M. Herman. RSTA on the move. In *Proc. DARPA Image Understanding Workshop*, pages 435–456, Monterey, CA, November 1994.

[Durić and Rosenfeld, 1995] Z. Durić and A. Rosenfeld. Stabilization of image sequences. Technical Report CAR-TR-778, Center for Automation Research, University of Maryland, College Park, 1995.

[Hansen *et al.*, 1994] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P.J. Burt. Real-time scene stabilization and mosaic construction. In *Proc. DARPA Image Understanding Workshop*, pages 457–465, Monterey, CA, November 1994.

[Horn, 1987] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4:629–642, 1987.

[Irani *et al.*, 1994] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 454–460, Seattle, WA, June 1994.

[Irani *et al.*, 1995] M. Irani, S. Hsu, and P. Anandan. Mosaic-based video compression. In *Proc. SPIE Conference on Electronic Imaging, Digital Image Compression: Algorithms and Techniques*, volume 2419, pages 242–253, San Jose, CA, 1995.

[Jazwinski, 1970] A. Jazwinski. *Processes and Filtering Theory.* Academic Press, New York, 1970.

[Kanatani, 1990] K. Kanatani. *Group-Theoretical Methods in Image Understanding.* Springer-Verlag, Berlin, 1990.

[Kwon *et al.*, 1995] O.J. Kwon, R. Chellappa, and C.H. Morimoto. Motion-compensated subband coding of video acquired from a moving platform. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2185–2188, Detroit, MI, January 1995.

Figure 8: 2D and 3D mosaics from 150 frames of a panning sequence. The top row shows the 70th and the last frame of the sequence. The second row shows the 2D mosaics constructed up to the corresponding frames, and the bottom row shows the corresponding mosaics using the 3D models and guessed camera parameters.

[Maybeck, 1982] P. Maybeck. *Stochastic Models, Estimation and Control.* Academic Press, New York, 1982.

[Morimoto and Chellappa, 1996] C.H. Morimoto and R. Chellappa. Fast electronic digital image stabilization. In *Proc. International Conference on Pattern Recognition*, volume 3, pages 284–288, Vienna, Austria, August 1996.

[Morimoto *et al.*, 1995] C.H. Morimoto, D. DeMenthon, L.S. Davis, R. Chellappa, and R. Nelson. Detection of independently moving objects in passive video. In I. Masaki, editor, *Proc. Intelligent Vehicles Workshop*, pages 270–275, Detroit, MI, September 1995.

[Morimoto *et al.*, 1996] C.H. Morimoto, P. Burlina, R. Chellappa, and Y.S. Yao. Performance analysis of model-based video coding. In *Proc. International Conference on Image Processing*, volume 3, pages 279–282, Lausanne, Switzerland, September 1996.

[Sawhney *et al.*, 1995] H. Sawhney, S. Ayer, and M. Gorkani. Model-based 2D and 3D dominant motion estimation for mosaicking and video representation. In *Proc. International Conference on Computer Vision*, pages 583–590, Cambridge, MA, June 1995.

[Tian and Huhns, 1986] Q. Tian and M.N. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics and Image Processing*, 35:220–233, 1986.

[Viéville *et al.*, 1993] T. Viéville, E. Clergue, and P.E.D.S. Facao. Computation of ego-motion and structure from visual and internal sensors using the vertical cue. In *Proc. International Conference on Computer Vision*, pages 591–598, Berlin, Germany, May 1993.

[Yao *et al.*, 1995] Y.S. Yao, P. Burlina, and R. Chellappa. Electronic image stabilization using multiple visual cues. In *Proc. International Conference on Image Processing*, volume 1, pages 191–194, Washington, D.C., October 1995.

[Young and Chellappa, 1990] G-S. J. Young and R. Chellappa. 3D motion estimation using a sequence of noisy stereo images: Models, estimation, and uniqueness results. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12:735–759, 1990.

[Zheng and Chellappa, 1993] Q. Zheng and R. Chellappa. A computational vision approach to image registration. *IEEE Trans. Image Processing*, 2:311–326, 1993.

Evaluation of Image Stabilization Algorithms

Carlos Morimoto **Rama Chellappa**
Center for Automation Research
University of Maryland, College Park, MD 20742-3275

Abstract

We evaluate the performance of several image stabilization algorithms using synthetic and real uncalibrated image sequences. Each algorithm is based on a different 2D parametric motion model, but they all share a similar structure. The basic algorithm estimates the inter-frame motion by fitting one of the transformation models using a feature-based multi-resolution technique. Stabilization is achieved by combining the inter-frame motion estimates with respect to a reference frame, and warping the current frame back to the reference. Results from several experiments that were carried out to evaluate the performance of each model are also presented. These experiments also demonstrate the influence of different system parameters, such as the use of multi-scale and subpixel feature tracking, on each model's overall behavior.

1 Introduction

Electronic image stabilization is the process of generating a compensated video sequence where unwanted camera motion is subtracted from the original input. Most proposed stabilization systems compensate for all motion [Burt and Anandan, 1994; Davis *et al.*, 1994; Hansen *et al.*, 1994; Irani *et al.*, 1994; Morimoto and Chellappa, 1996; Sawhney *et al.*, 1995], producing a sequence where the background remains motionless.

Since motion estimation is the main component of an image stabilization system, the evaluation of the system could be based on the performance of the motion estimation module alone, in which case one could use synthetic or calibrated sequences where the inter-frame motions are known, such as in [Barron *et al.*, 1994]. Aside from the issue of generating sequences with known motion, most stabilization systems use approximate parametric global transformations, which creates the problem of finding the optimal transformation from the ground truth data, so that the motion estimates can be evaluated in terms of a distance measure from these optimal parameters. Another important issue is how to compare the performance of systems based on different motion models, since distance measures might be model-dependent.

Other methods of evaluating image stabilization systems are presented in [Balakirsky and Chellappa, 1996; Morimoto and Chellappa, 1996]. [Balakirsky and Chellappa, 1996] compares the performance of different stabilization algorithms based on the accuracy of a real-time object tracker, and [Morimoto and Chellappa, 1996] considers the maximum displacement velocity in pixels per second, computed as the product of the frame rate and the maximum image displacement between frames.

In this paper, we evaluate the fidelity of image stabilization techniques using the power signal-to-noise ratio (PSNR) between stabilized frames. This method does not require the use of calibrated sequences to compare different systems, and provides a simple way of comparing systems based on different motion models. Synthetic sequences are used to measure other system properties, such as the range of displacements within which they operate.

Intuitively, since all motion is compensated for after stabilization, the difference between two stabilized frames should be, ideally, zero in the overlapping regions. Several factors contribute to this measure not being zero. For stabilization purposes, the PSNR can be considered as a measure of the departure from the optimal case, or as a measure of the overlap between two images, which is maximized when the

The support of the Defense Advanced Research Projects Agency (ARPA Order No. A422) and the U.S. Army Research Office under Grant DAAH04-93-G-0419 is gratefully acknowledged

images are identical. When two images do not overlap, stabilization is not possible, and the PSNR is meaningless. But if pixels from non-overlapping regions are replaced by the corresponding pixels from the original frame before the PSNR is computed, a lower bound (LB) is created for the fidelity measure, which is given by the PSNR between the corresponding frames from the original sequence; and we assume that the stabilization system has produced a valid output whenever the PSNR is higher than LB. Erroneous motion estimates can in fact produce PSNRs below LB.

In the following section we present a brief description of the stabilization system used in our evaluation experiments. These experiments compare the performance of several systems based on different transformations for motion estimation and compensation, and also demonstrate the influence of different system characteristics, such as multi-resolution estimation and subpixel feature tracking, on the system's overall behavior. Section 3 presents three experiments to evaluate and compare some properties of image stabilization systems. The results of each experiment are presented and discussed in Section 4. Section 5 concludes the paper.

2 Image Stabilization Algorithm

The algorithms used in this paper are based on the 2D fast image stabilization system described in [Morimoto and Chellappa, 1996], extended to other subsets of the group of affine image transformations. The system can be decomposed into three main modules, as shown in Figure 1. From the video input, the first module estimates the motion between two consecutive frames using a multi-resolution feature-based technique. The motion compensation module uses the inter-frame estimates to update the global transformation used to bring the current frame into alignment with the reference frame. The third module uses the global transformation to generate the stabilized output sequence, and possibly mosaics.

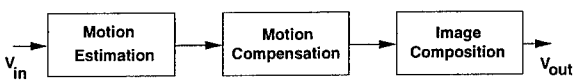


Figure 1: Modules of a general electronic image stabilization system.

Instead of considering the problem of estimating and compensating for the camera motion to generate a stabilized sequence as in [Morimoto and Chellappa, 1996], we consider stabilization as an image registration problem, where an image I_0 is mapped into an image I_1 according to [Brown, 1992]

$$I_1(x, y) = \gamma(I_0(\psi(x, y))) \quad (1)$$

where $I_i(x, y)$ represents the intensity of pixel (x, y) of image I_i, γ is a one-dimensional intensity transformation function, and $\psi(x, y)$ is a two-dimensional spatial coordinate transformation which maps pixel coordinates $\mathbf{p}_0 = (x_0, y_0)$, to new pixel coordinates $\mathbf{p}'_0 = (x'_0, y'_0)$ such that $\mathbf{p}'_0 = \psi(\mathbf{p}_0)$. For stabilization purposes, this problem can be reduced to the computation of the optimal spatial transformation ψ which properly aligns the two frames.

2.1 Coordinate Transformations

To facilitate the estimation of the coordinate transformations between frames, their composition, and other operations necessary for stabilization, we will restrict our experiments to subsets of the group of affine transformations.

A simple transformation which is sufficient to register two images taken from the same viewing angle but from a different position can be defined using four parameters such as

$$\mathbf{p}_1 = s\mathbf{R}\mathbf{p}_0 + \mathbf{T} \quad (2)$$

where $\mathbf{T}$ is a translation vector, s is a scalar corresponding to the scaling factor, and $\mathbf{R}$ is an orthogonal rotational matrix defined by

$$\mathbf{R} = \begin{pmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{pmatrix} \quad (3)$$

where the parameter Θ defines a rotation around the viewing axis. These transformations preserve angles and relative lengths, and belong to the *similarity* group.

Rigid transformations where scaling is not allowed are also described by (2) when s is set to be of unit value. The set of all such 3-parameter transformations forms the *Euclidean* group.

A 6-parameter *affine* transformation is obtained by relaxing the constraints on the rotational matrix $\mathbf{R}$; it is defined by

$$\mathbf{p}_1 = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \mathbf{p}_0 + \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \mathbf{R}\mathbf{p}_0 + \mathbf{T} \quad (4)$$

Angles and lengths are no longer preserved in this transformation, although parallel lines remain parallel. The affine transformation allows for skewing, and for change in aspect ratio due to non-uniform scaling in x and y.

The use of higher-order models is discussed in [Mann and Picard, 1995]. The family of transformations described so far cannot account for some distortions which appear in more general 3D motion,

such as those caused by pan-tilt movement. The 8-parameter transformations of the *projective* group, which are able to describe general 3D camera motion, require more complex estimation techniques which considerably reduce the speed of the stabilization system.

2.2 Parameter Estimation

In order to compute the coefficients of the transformations described above, a multi-resolution estimation technique similar to the one described in [Zheng and Chellappa, 1993] is used.

Initially, Gaussian and Laplacian pyramids are constructed for both of the images to be registered, I_t and I_{t-1}. A Gaussian pyramid G_t is formed by combining several reduced-resolution Gaussian images of I_t. The image at level l of the pyramid is denoted by $G_t[l]$, where $G_t[0]$ is the highest resolution image, which might be a copy of I_t. An image at level l has resolution $R[l] = R[0]/2^l$, where $R[0]$ is the resolution of $G_t[0]$. The Laplacian pyramid L_t is obtained by convolving G_t with a Laplacian kernel operator. The levels of the pyramids G_{t-1} and G_t are used from coarse to fine resolution; each new processed level contributes to refining the motion estimates.

Starting from the coarsest level c, N features are chosen by dividing $L_{t-1}[c]$ into N non-overlapping regions and selecting the pixel with maximum intensity value in each of these regions for tracking. A match for the corresponding feature from $G_{t-1}[c]$ is obtained by minimizing the sum of squared differences (SSD) over a neighborhood (search window) around the candidate matches in $G_t[c]$.

For a feature $G_{t-1}[c](x, y)$, a search for the minimum SSD is performed in a window of size $S = (2s+1)\times(2s+1)$ centered at the pixel $G_t[c](x, y)$. After the grid-to-grid matches are obtained, displacements with subpixel accuracy are computed using a differential method [Tian and Huhns, 1986]. Subpixel accuracy is necessary to eliminate the quantization error introduced when the images are digitized.

The transformation parameters are computed from the feature displacements as follows. Each tracked feature contributes two equations from the x and y coordinates of (2) and (4). Since in general we have $2N > P$, where P is the number of parameters, the over-determined system with $2N$ equations and P unknowns can be solved using a least-square method.

For an arbitrary higher resolution level l, the transformation estimated up to level $l+1$ must be properly scaled to level l and used to warp $G_t[l]$. The registration process continues by scaling the features from $L_{t-1}[l+1]$ or computing new features in $L_{t-1}[l]$, and tracking the features from $L_{t-1}[l+l]$ to the warped image of $L_t[l]$. The transformation computed from the feature displacements at level l must be combined with the estimate from the previous level to produce the correct inter-frame transformation used to warp $G_t[l-1]$. This process is repeated until the finest resolution level is reached. Notice that since the displacement is doubled after every level, the total displacement that this algorithm can handle can be very large even for small search window sizes.

2.3 Motion Compensation

To generate a stabilized sequence it is necessary to determine the transformation which brings the current frame into alignment with the reference frame. The motion compensation module computes this global transformation from the inter-frame estimates. Let ψ_t be the global transformation which aligns image frame I_t with the reference frame I_0, i.e., $I_0(x, y) = I_t(\psi_t(x, y))$. When the inter-frame estimate ψ which aligns I_{t+1} with I_t is available ($I_t(x, y) = I_{t+1}(\psi(x, y))$), the new global transformation ψ_{t+1} must be updated using the composition rule

$$\psi_{t+1}(x, y) = \psi_t(\psi(x, y)). \tag{5}$$

2.4 Image Composition

The stabilized output sequence is generated by warping the current frame according to the global transformation computed by the motion compensation module. Warping is performed by scanning the output frame and determining the intensity at the corresponding pixel of the input frame, which is then copied onto the output in case the transformed pixel lies inside the input image; otherwise, some background default intensity is placed at the current output pixel. In general, the transformed coordinates will have non-integer values. To obtain the intensity at this non-grid point $(x, y)' = \psi(x, y)$, bilinear interpolation using the nearest four grid points (NW, NE, SW, SE) is performed

$$\begin{aligned} &\mathrm{Out}(x, y) = \\ &\mathrm{NW} \times dx_2 \times dy_2 + \mathrm{SW} \times dx_2 \times dy_1 + \\ &\mathrm{NE} \times dx_1 \times dy_2 + \mathrm{SE} \times dx_1 \times dy_1 \end{aligned} \tag{6}$$

where dx_1, dx_2, dy_1, and dy_2 are shown in Figure 2.

3 Evaluation Tests

The following experiments were designed to evaluate two characteristics of stabilization systems: fidelity and displacement range. *Fidelity* is a measure of how well the stabilization is compensating the motion of the camera, i.e., how precisely the motion model fits

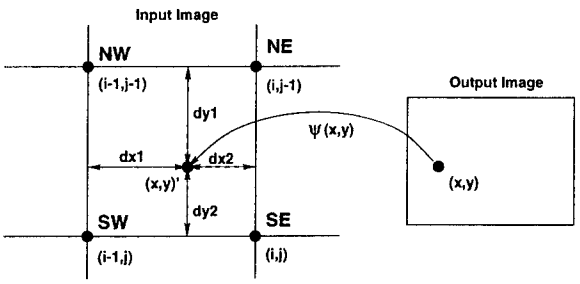


Figure 2: Bilinear interpolation.

the actual camera motion. Since the motion must be estimated between frames, it is also directly dependent on the estimation process. *Displacement range* is defined by the minimum and maximum image displacements which can be estimated. The displacement range is a key feature of a stabilization system which, combined with the frame rate, determines the range of image velocities, in pixels per second, which can be compensated for.

A brief remark must be made here about algorithm complexity. Since most algorithms are targeted for real-time hardware platforms, solutions of higher computational complexity may be more appropriate than solutions of lower complexity. Since we are interested in overall performance, we concentrate on empirical, possibly qualitative ways of comparing different systems.

3.1 Experiment 1: Fidelity

The PSNR between stabilized frames can be used to measure the fidelity of a system. Intuitively, when all motion is compensated for, there should be no residual motion after stabilization, which means that the same frame should be obtained over and over again. Since the images are the same, the difference between two stabilized images should be zero for every pixel. Many factors contribute to this difference being non-zero, such as noise, estimation errors, distortions caused by departures from the motion model and by the interpolation during warping, etc. We want fidelity to measure the imprecision of the system due to all these factors.

The MSE is a measure of the average departure per pixel from the desired stabilized result. The PSNR between I_1 and I_0 is

$$PSNR(I_1, I_0) = 10 \log \frac{255^2}{MSE(I_1, I_0)} \qquad (7)$$

The PSNR gives a relation between the desired output and the residual image, in terms of their powers (for gray images with a maximum intensity of 255). The higher the PSNR between two stabilized frames, the better the fidelity of the system.

The above formulation does not account for the fact that, when a camera moves, it probably produces non-overlapping regions where compensation cannot be done. If the PSNR were computed just for the overlapping areas, the fidelity measure would not be meaningful when the overlapping areas are small. In order to handle these regions, we propose that every pixel belonging to a compensated frame and which does not overlap with the reference frame be copied from the current frame before computing the PSNR. For the case when the motion estimate warps the image completely outside the reference frame, we have a natural lower bound (LB) which is given by the PSNR between the reference and the current frames without compensation. We assume that the system has produced a valid output whenever the PSNR between stabilized frames is higher than LB.

To run this experiment, the warping functions were modified to account for the non-overlapping areas. Given a particular stabilization system and an arbitrary sequence, two measures are computed. The first is a measure of the inter-frame transformation fidelity (ITF), and the second measures the global transformation fidelity (GTF). ITF is defined as the PSNR between two consecutive stabilized frames, and GTF corresponds to the PSNR between the reference frame and the current stabilized frame. The lower bounds on ITF and GTF will be respectively denoted by LB_i and LB_g.

3.2 Experiment 2: Minimum Image Displacement

The second experiment determines the minimum image displacement that a system can measure. Since the estimation is based on feature tracking, this experiment could also be used to compare different tracking algorithms with subpixel precision. For this experiment, synthetic image sequences were created by the following procedure. Given a single image I of large dimensions, a window of smaller size (e.g. 128×128) is first placed at a fixed position on the image. This window is used to compose the output sequence. The first frame is defined by the window itself, and the displacement velocity increment (acceleration) is set to zero. The following frames are created by incrementing the displacement velocity by a small amount, and warping I according to the new displacement velocity using bilinear interpolation. As a result, the contents of the window, when placed on the warped image, change proportionally. The precision of this measurement is defined by the acceleration step between frames.

For very small displacements, the PSNR between consecutive frames is very high, i.e., LB_i is very high. If the errors in the estimated parameters are big-

ger than the true transformation parameters, ITF is lower than LB_i. As the displacement increases, LB_i decreases and ITF increases if the displacement is large enough to be estimated. Therefore, one curve eventually crosses the other. This crossing point is used to define the minimum image displacement for which the system can compensate.

3.3 Experiment 3: Maximum Image Displacement

This experiment determines the upper bound on the range of displacements that can be handled by a system. Synthetic image sequences were created using the method described above, using larger acceleration steps between frames.

It is expected that when the system is working properly, ITF remains higher than LB_i, which is low for large inter-frame displacements. When the displacement is too large, the system produces invalid motion estimates, causing ITF to drop and possibly become lower than LB_i. We define the maximum image displacement to be the point where the ITF curve crosses the LB_i curve.

4 Experimental Results

All experiments were run using the same settings for all parameters, i.e., the same number of feature points, the same number of pyramid levels, the same search window sizes, etc. 16 features arranged in two rows and eight columns were tracked using search windows of size 5 pixels per pyramid level. Two pyramid levels were constructed for images of resolution 128×128, and three levels were used for images of higher resolution. All synthetic sequences were of resolution 128×128, and all real uncalibrated sequences were of resolution 320×240.

In the graphs presented in this section, the following notation is used. The curves for the affine transformations are drawn using dotted lines with stars (.*.); the curves for the similarity transformations are drawn using solid lines (-); and the curves for Euclidean transformations, with dash-dotted lines (-.). Dotted lines with circles (.o.) are used for the measurements' lower bounds, and other curves (if any) are drawn with dotted lines with crosses (.+.).

Figure 3 shows the results of evaluating the three systems using two uncalibrated sequences. The first column shows the results for the UGV sequence, which is composed of 30 frames of real video, where the camera starts zooming out and than pans from right to left. The graph on top of the first column shows ITF and LB_i for all frames. Observe that the affine- and similarity-based systems have very similar curves, while the Euclidean system performs poorly during the first ten frames, which correspond to the zooming part of the sequence. This result is expected since the Euclidean group does not model scaling.

After the 20th frame, the sequence does not overlap the reference frame. This can be observed from the GTF curves shown in the bottom graph in the first column of Figure 3. The GTF drops from frame to frame since each new frame has less overlap with the reference frame. The GTF of the Euclidean system is considerably smaller due to the lack of scaling compensation.

The second column of Figure 3 shows the ITF and GTF for the Building sequence. This sequence is also composed of 30 frames of real video, and contains a simple pan from left to right. In this case, since there is no change in scale, all the curves are very similar, i.e., all systems perform about equally well. It is important to notice from the ITF graph that feature outliers have much more influence on the performance of higher-order models. To test this hypothesis, the affine system was reconfigured to use 20 features instead of 16 (shown by the (.+.) curve); the resulting performance improvement can be seen from the ITF and GTF curves. Since the same 16 features are used for all systems, the least-square fit seems to be much more robust for the lower-complexity models. Both the UGV and Building sequences are available in MPEG format at http://www.cfar.umd.edu/~carlos/IUW97EVALUATION.html.

Figure 4 shows the results of determining the minimum displacement for each system. Two synthetic sequences composed of 19 frames each were created for this experiment. The inter-frame acceleration step was set to 1/10th of a pixel/frame^2, i.e., frame F_n has a displacement of $n/10$ pixel from F_{n-1}, for $n > 0$. The original (Bahia and Boat) sequences are available at the same http address. For this experiment we introduced a fourth system based on the Euclidean group, but with a simpler grid-to-grid (no subpixel precision) feature tracker. The measurements for this system are shown by dotted lines with crosses (.+.). For the first three systems, ITF becomes larger than BL_i after the second frame, i.e., the minimum displacement is below 0.3 pixel/frame. For the new system, the minimum displacement is below 0.6 pixel/frame for the Boat sequence, and below 0.7 pixel/frame for the Bahia sequence.

Figure 5 shows the results of determining the maximum displacements. Similar sequences were created, now with an acceleration of 1 pixel/frame^2. The maximum displacement is a property of the system related to the search sizes and number of pyramid levels. The search sizes were set to ± 5 pixels on each

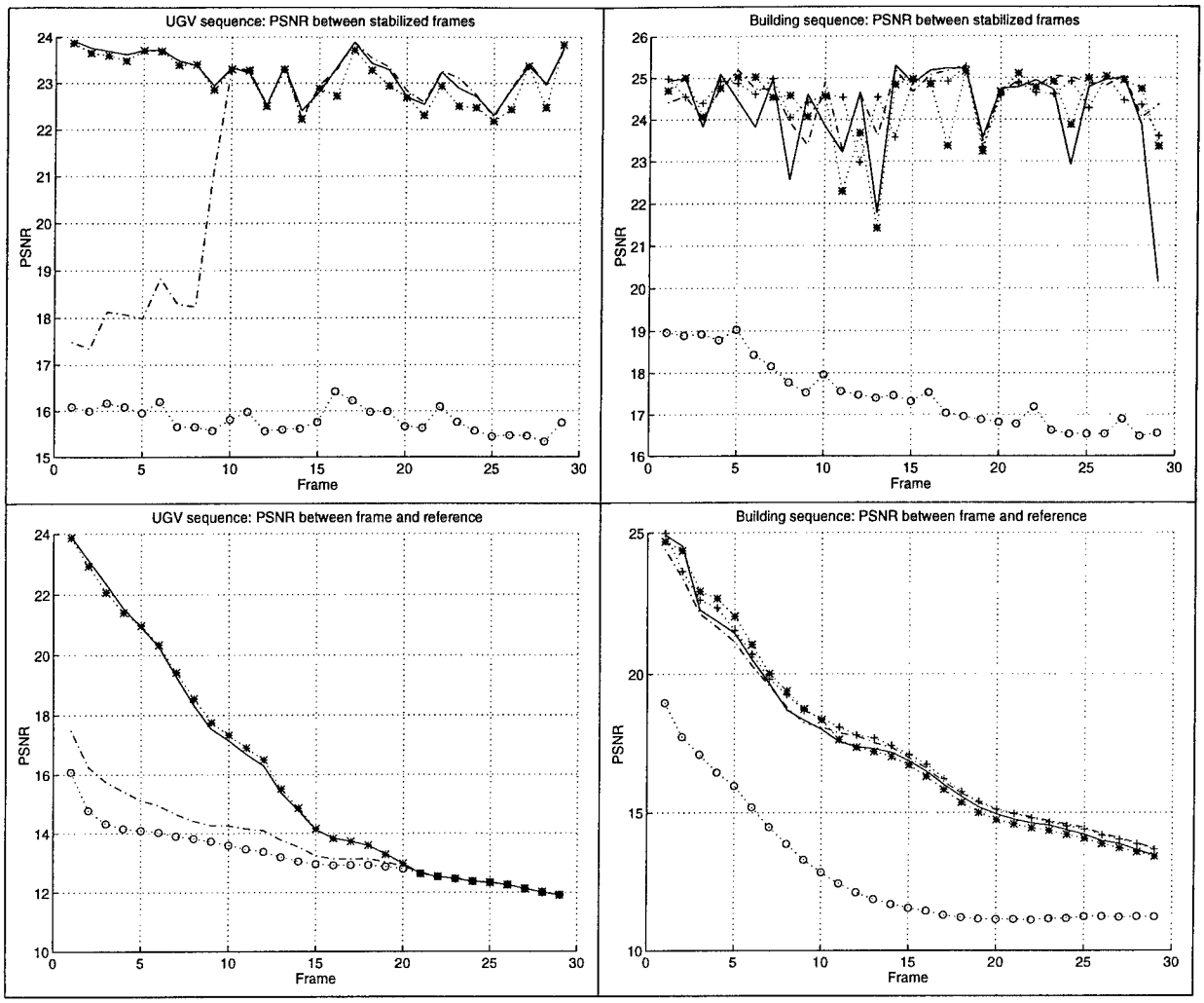


Figure 3: Results from Experiment 1. The (.*.) curve shows the results for the affine fit, (.+.) for affine using 20 feature points, (-) for similarity, (-.) for Euclidean, and (.o.) for lower bound.

level, and two pyramid levels were constructed. A fourth system with the same search window sizes but only one pyramid level (without multi-resolution) was also tested; its results are shown by the (.+.) curves.

For the Boat sequence, the maximum displacement of the first three systems lies between the 13th and 14th frame, so that it is safe to say that it is above 13 pixels. For the Bahia sequence, which is of lower quality, this also seems to be the case, although there is a considerable drop after the 10th frame. For the system using only one pyramid level, the maximum displacement lies around 5 pixels, which corresponds to the size of the search window.

The analysis of minimum and maximum displacements is not limited to translations. It is simple to create a synthetic sequence by varying any parameter of a transformation group, and then empirically determining the system's operating range for the sequence. For example, Figure 6 shows the lower and upper bounds for rotation sequences created by varying the rotational parameter of the Euclidean transformation. The top graph of Figure 6 shows the results for determination of the minimum rotation. The sequence, based on the Bahia image, was created using rotation increments of 0.1 degree. The minimum rotation for all systems lies below 0.3 degree. The bottom graph of Figure 6 shows the results for determination of the maximum rotation. A second sequence was created using rotation increments of 1 degree, but starting from 5 degrees. The ITF of all systems seems to break down after the 19th frame, i.e., the maximum rotation is above 23 degrees.

5 Conclusion

We have proposed a simple way of evaluating the fidelity and range of displacements of stabilization systems. Fidelity is a measurement of how good the

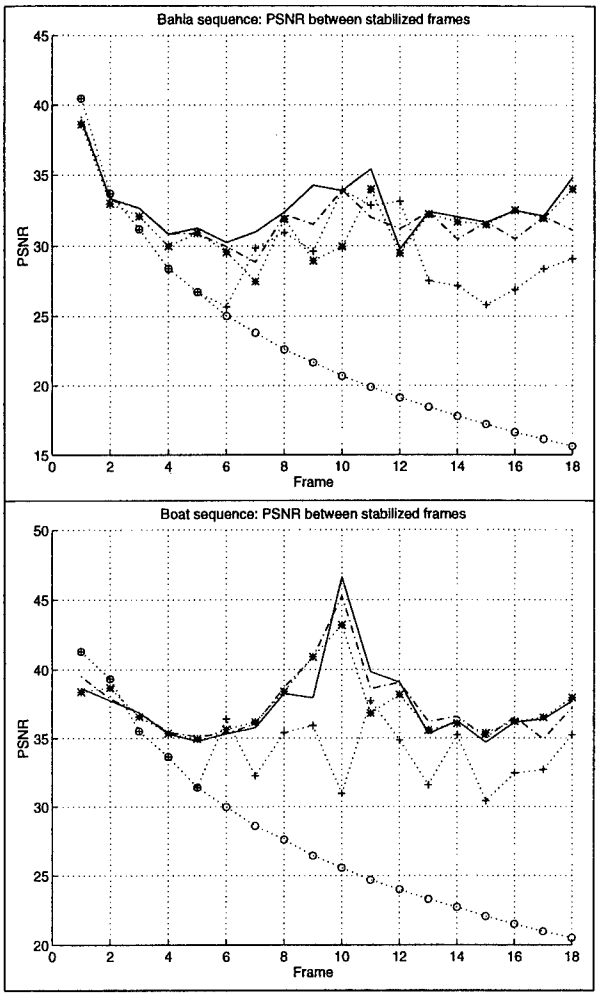


Figure 4: Results from Experiment 2 - determination of minimum translational displacement. The (.*.) curve shows the results for the affine fit, (-) for similarity, (.+.) for Euclidean without subpixel precision, (-.) for Euclidean, and (.o.) for lower bound.

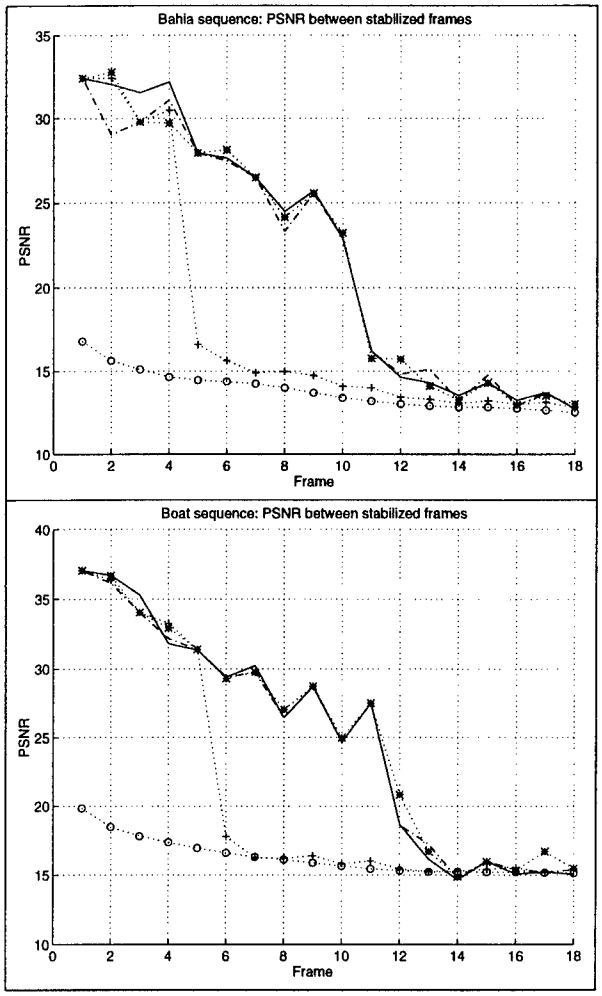


Figure 5: Results from Experiment 3 - determination of maximum translational displacement. The (.*.) curve shows the results for the affine fit, (-) for similarity, (.+.) for Euclidean without multiresolution, (-.) for Euclidean, and (.o.) for lower bound.

estimated image transformation fits the true transformation, and the range of displacements characterizes the minimum and maximum displacements that a system can handle. Although these measurements are not absolute since they depend on the sequence being stabilized, they can be used to compare different systems, even those based on different transformation models. They can also be used as development tools, to easily compare performance as a function of different system parameters and modules.

Uncalibrated sequences can be used to compare the fidelity of systems; synthetic ones are required to measure the range of displacements. We have evaluated the performance of stabilization systems based on three different transformation groups, the Euclidean, similarity, and affine groups. The experimental results seem to prove Occam's razor: Applying more complex models to fit the data does not necessarily produce better results. Actually, it turns out that the more complex models are more sensitive to tracking errors, causing them to perform worse than the simpler models. We verified that increasing the number of features lowers the difference between the systems' performances, and a significant number of features can actually make the more complex models perform better.

The range of displacements is another key feature for the evaluation of stabilization systems. Suppose the minimum and maximum displacements are 0.5 and 10 pixels/frame respectively, and the system operates at 10 frames per second. This means that if the velocities of the images being stabilized are lower than 5 or higher than 100 pixels/second, the system will not operate properly.

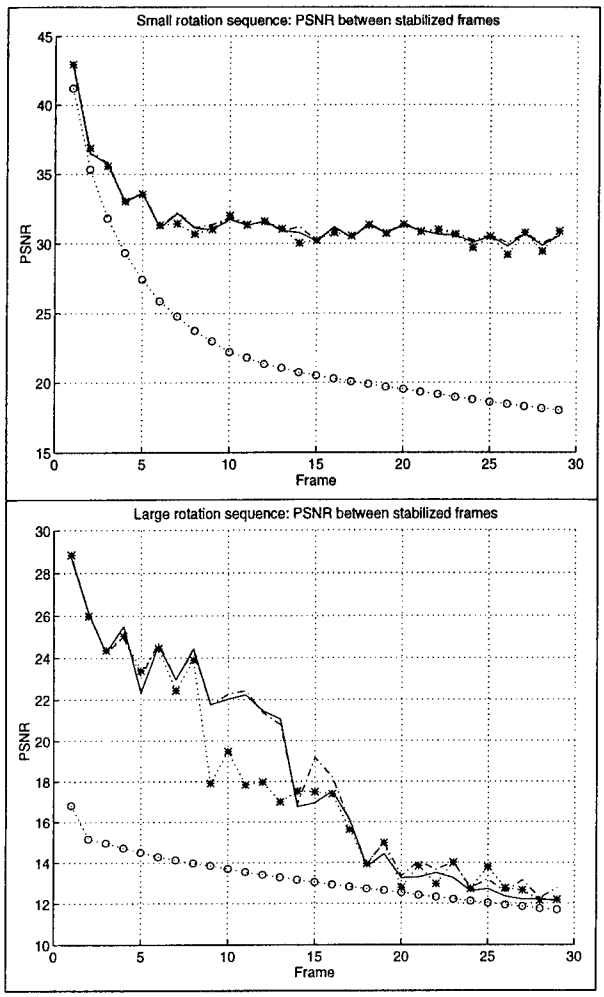


Figure 6: Determination of the rotation estimation range. The (.*.) curve shows the results for the affine fit, (-) for similarity, (-.) for Euclidean, and (.o.) for lower bound.

References

[Balakirsky and Chellappa, 1996] S. Balakirsky and R. Chellappa. Performance characterization of image stabilization algorithms. Technical Report CAR-TR-822, Center for Automation Research, University of Maryland, April 1996.

[Barron *et al.*, 1994] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77, 1994.

[Brown, 1992] L.G. Brown. A survey of image registration techniques. *Computing Surveys*, 24:325–376, 1992.

[Burt and Anandan, 1994]
P. Burt and P. Anandan. Image stabilization by registration to a reference mosaic. In *Proc. DARPA Image Understanding Workshop*, pages 425–434, Monterey, CA, November 1994.

[Davis *et al.*, 1994] L.S. Davis, R. Bajcsy, R. Nelson, and M. Herman. RSTA on the move. In *Proc. DARPA Image Understanding Workshop*, pages 435–456, Monterey, CA, November 1994.

[Hansen *et al.*, 1994]
M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P.J. Burt. Real-time scene stabilization and mosaic construction. In *Proc. DARPA Image Understanding Workshop*, pages 457–465, Monterey, CA, November 1994.

[Irani *et al.*, 1994] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 454–460, Seattle, WA, June 1994.

[Mann and Picard, 1995] S. Mann and R.W. Picard. Video orbits: Characterizing the coordinate transformation between two images using the projective group. Technical Report 278, MIT Media Lab, 1995.

[Morimoto and Chellappa, 1996]
C.H. Morimoto and R. Chellappa. Fast electronic digital image stabilization. In *Proc. International Conference on Pattern Recognition*, volume 3, pages 284–288, Vienna, Austria, August 1996.

[Sawhney *et al.*, 1995] H. Sawhney, S. Ayer, and M. Gorkani. Model-based 2D and 3D dominant motion estimation for mosaicing and video representation. In *Proc. International Conference on Computer Vision*, pages 583–590, Cambridge, MA, June 1995.

[Tian and Huhns, 1986] Q. Tian and M.N. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics and Image Processing*, 35:220–233, 1986.

[Zheng and Chellappa, 1993]
Q. Zheng and R. Chellappa. A computational vision approach to image registration. *IEEE Trans. on Image Processing*, 2:311–326, 1993.

Multiple Perspective Interactive Video Surveillance and Monitoring

Jeffrey Boyd **Edward Hunter** **Ramesh Jain** **Patrick Kelly**
Jennifer Schlenzig **Andy Tai**
Visual Computing Laboratory, Electrical and Computer Engineering
University of California at San Diego, La Jolla, CA 92093-0407
E-MAIL: jeffboyd,ehunter,jain,phkelly,schlenz,atai@ece.ucsd.edu
HOMEPAGE: http://vision.ucsd.edu/

Abstract

Multiple perspective interactive video (*MPI-Video*) is an infrastructure for the analysis, management and interactive access to multiple streams of video cameras monitoring a dynamically evolving scene. Two important concepts form a basis for *MPI-Video*. The first, content-based interactivity, allows a user of a system to access information based on content and context, thus allowing the user to retrieve useful information even when the volume of data available is large. The second, gestalt perception, refers the merging of data from multiple image sensors into a single percept that conveys more information than just the individual images. This paper describes current research on application of *MPI-Video* to video surveillance and monitoring. Work focuses on the assimilation of multiple streams of video into a single, integrated representation of the world. This forms the basis for future work to build a database subsystem to support content-based query operations, and a query environment which allows navigation and querying of a wealth of data. We describe research that we are pursuing to assimilate data for *MPI-Video*. This includes assimilation of motion and optical flow, determination of kinematic structure of objects, recognition of activities, and three-dimensional visualization.

1 Introduction

The Multiple Perspective Interactive Video (MPI-Video) project has been active for more than two years and has already demonstrated its applicability in areas including video surveillance and monitoring. *MPI-Video* is an infrastructure for the analysis, management and interactive access to multiple streams of video cameras monitoring a dynamically evolving scene. Multiple Perspective Interactive Video [Jain and Wakimoto, 1995, Kelly *et al.*, 1995], *MPI-Video*, provides a framework for the management of and interactive access to multiple streams of video data capturing different perspectives of related events. *MPI-Video* has dominant database and hyper-media components which allow a user not only to interact with live events, but browse the underlying database for similar or related events. The interactive construction of queries is also supported.

For video surveillance and monitoring (VSAM) large areas, sensor data from many platforms must be analyzed in a unified manner. Since battlefields or any important urban site are too large to be covered just by one camera, it is essential that multiple platforms be used to acquire data from multiple perspectives. This system should be operational, independent of the time of the day and the season. This will require different types of sensors. The system of all these sensors mounted on multiple platforms should function in unison and present a *Gestalt view* to a user. Important research issues that must be addressed in this area include, assimilation of information from multiple sensors, determination of camera placement,

dynamic scene segmentation, event understanding, camera hand-off, and representation of individual sensor and global information.

In this paper, first we define two very important concepts, Content-based Interactivity and Gestalt Perception that are central to the *MPI-Video* infrastructure, and then discuss *MPI-Video* infrastructure in the context of VSAM application. Finally, we discuss current research in related areas of image understanding in our laboratory.

2 Content-based Interactivity

Interactive TV and Video-on-demand have been often talked about features of multimedia systems. Interactivity is very attractive; the current popularity of virtual reality and cyber-communities, including chat rooms, can be largely attributed to their strong interactive component.

A major limitation of current TV, video, and movies is their passive one-way flow of information. Users have no control over the content and how they can view it. A powerful interactive environment can give a viewer a feeling of being present at the event being observed and view the objects and events of interest. It will be soon possible to provide highly interactive real immersive environments that will provide a user a feeling of telepresence. In fact, it will be possible that in addition to the feeling of being present, for many events users will be able to extract information of interest and see other information related to events, like scenes from other movies by the same actor, or similar moves by a particular player in other games.

A good example of flexible interactive environments are video games. Though a player is immersed in 'virtual environment', he has freedom to interact with environment through his avatar. It is this interactivity that makes video games so popular. Even in their early days when the quality of graphics was very primitive, interactivity popularized and carried video games. Compare the interactivity offered by video-on-demand and interactive movies. Clearly the interactivity offered in these systems is limited to very simple 'branching' condition at fixed points, whereas the interactivity offered in video games gives one freedom to act at any point in time and space; of course, depending on the context, some constraints are imposed and the results of actions depend on the context.

When the amount of information grows, human ability to remember correct information sources becomes overloaded and starts failing. The success of databases is due to their ability to allow access to the content of the databases based on the queries related to the content. On the world wide web, search engines have played a major role in easier access to textual information. Currently, commercial tools to provide content-based address to visual information are in their infancy.

A video or a television event is a vast stream of data representing intensity values at points in an image. This intensity value represents some physical attributes in space for the scene captured by a camera. Viewers are interested in objects, their characteristics, relationships, and temporal history. A video is interesting because it provides that information.

We can also view a physical event as an evolution of spatio-temporal characteristics at a certain location. As the amount of data increases, human ability to specify the locations decreases. Thus, a system that will provide facilities to specify objects and events and will return or retrieve corresponding data will be much more interesting and useful to humans. Content-based interactivity is not only desirable, in systems with large volume of data, content-based interactivity is essential.

3 Gestalt Perception

At any given time, we can only see the world, or the environment, from one perspective. To get other perspectives, we must move our eyes. To explore the environment from other viewpoints, we have to physically move. When we view the environment from one perspective, we are limited to what one may call tunnel vision or more precisely, considering the nature of image formation process, funnel vision. This was

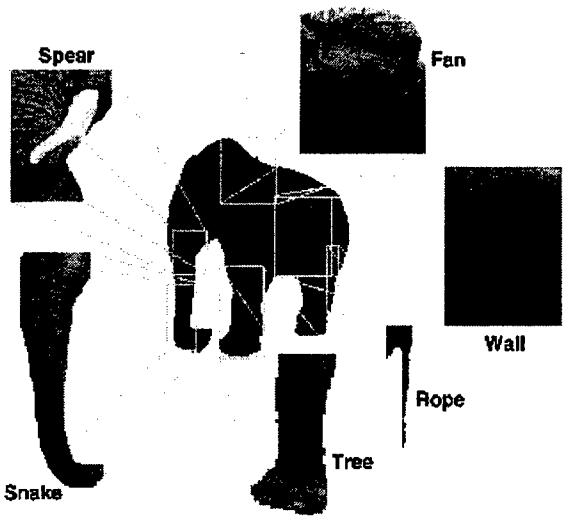


Figure 1: The fable of six blind men and an elephant. Each man perceives only a small window into reality. If the six men were not blind, but looked at the elephant with zoomed-in cameras, they would still make the same mis-classifications.

realized long time ago, as is clear from the famous fable of Six Blind Men and an Elephant (see Figure 1). Cameras have similar limitations; a camera captures a scene from a limited perspective.

One could obtain more information about the environment by panning and tilting the camera so that one could see a complete view from one position. Quicktime VR has attracted attention by providing a mechanism to record a scene from one position and then allowing a user to view the scene along any direction. Similar efforts are being made in many research groups by taking multiple images of a scene and then using software to merge these images to provide a larger picture than is possible from any single camera views.

As we show below, using a powerful information system to mediate between viewers and multiple cameras, it is possible to provide gestalt vision, which is more than any individual camera. A viewer than can see the scene from any position and may walk through a dynamic scene without disturbing the events in the scene.

4 MPI Video

The *MPI-Video* system provides an infrastructure for analysis of video data from multiple cameras viewing a common area, and integrates this information into a dynamically evolving database to support content based retrieval activities. Thus, operations in an *MPI-Video* system run the gamut from low-level analysis executed on the video frames themselves to more high-level data modeling, storage, retrieval and indexing operations. The *MPI-Video* environment is a heterogeneous, distributed information infra-structure. Thus, it possesses the very qualities that a successful VSAM architecture must have. The primary source of information is a number of live video streams acquired from a set of cameras covering a closed environment. This environment has a static component consisting of a model of the environment which resides on a server. The server also contains a library of possible dynamic objects that can appear in the environment. Multiple sensors capture the event and the system dynamically reconstructs a sequence of camera-independent three-dimensional scenes from the video streams [Katkere *et al.*, 1997]. In *MPI-Video*, the user can ask questions, specify alarming activities, and view and navigate in this world as the real-life event unfolds. While remaining in this world, the user may also request additional information on any static or dynamic object. Secondary information resources such as hyper-linked HTML documents, databases of static images, and ftp sites of reference archives are available to the system and may need to be accessed either to initiate a user query or as the result of a query.

The *MPI-Video* architecture shown in Figure 2 [Jain and Wakimoto, 1995, Kelly *et al.*, 1995] has the following components:

1. **Video Data Analyzer:** The *MPI-Video* system must detect and recognize objects of potential interest and their locations in the scene. This requires powerful image segmentation methods.

2. **Environment Model Builder:** Individual camera scenes will be combined in this

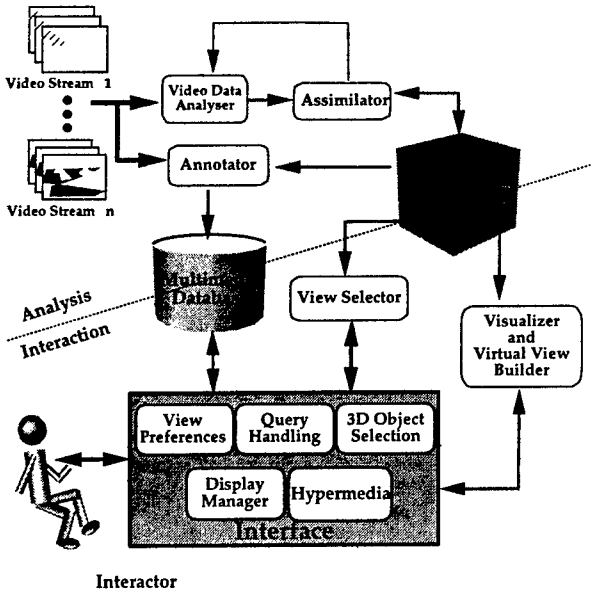


Figure 2: *MPI-Video* system architecture overview.

system to form a model of the environment. All potential objects of interest and their locations will be recorded in the environment model. The representation of the environment model depends on the application domain and the facilities provided to the viewer.

3. **Viewer Interface:** A viewer is able to select the perspective that he or she desires. This information should be obtained from the user in a friendly but directed manner.

4. **View Selector:** The view selector responds to the user's request by selecting appropriate images to be displayed. These images may all come from one perspective or the system may have to select the best camera at every point in time to display the selected view and perspective.

5. **MPI-Video Database:** If the event is not a real time event, then it is possible to store the episode in an *MPI-Video* database. Each camera sequence will be stored along with its meta-data. Some of the meta-data is feature based and allows content-based operations [Jain and Hampapur, 1994, Swanberg *et al.*, 1993]. Data can also be collected during a real time event and stored for later use.

6. **Virtual View Builder:** A particularly important component of *MPI-Video* is *Immersive Video* [Moezzi *et al.*, 1996], where a *virtual camera* is created for the viewer by combining the extracted model with the original video streams to give a sense of omniscient presence. The viewer in an *Immersive Video* environment is no longer controlled by the limitations of a physical camera.

Our current research has focused on implementation and evaluation of the MPI Video architecture outlined in Figure 2. Figure 3 shows our current MPI Video interface. Our current system has a variety of features that are necessary in a VSAM system, for instance, a graphical model of the surveilled environment. Display of appropriate video streams (in this case the environment is monitored by a total of six cameras). The current system also supports the interactive identification of a particular area to be monitored. When objects enter the monitored area, it is flagged as a visual alarm for the user. A simple environment model maintains information about the objects currently in the environment, in this case, their identities, presented to the user in the text list and their locations graphically indicated in the model. Users can select objects on the list or in the model to retrieve information about the selected object. We must now focus on refining the components, the Environment Model being the key among them.

Three aspects central to this architecture are[Kelly *et al.*, 1995]:

1. Video data analysis and the assimilation of the multiple streams to form a single, integrated world-representation. Selection of a "best view" from the input data stream.

2. A database subsystem which stores the raw video data, the derived data generated by the video analysis portion and any meta-data input by the user. The database supports content-based query operations by the user or software agents.

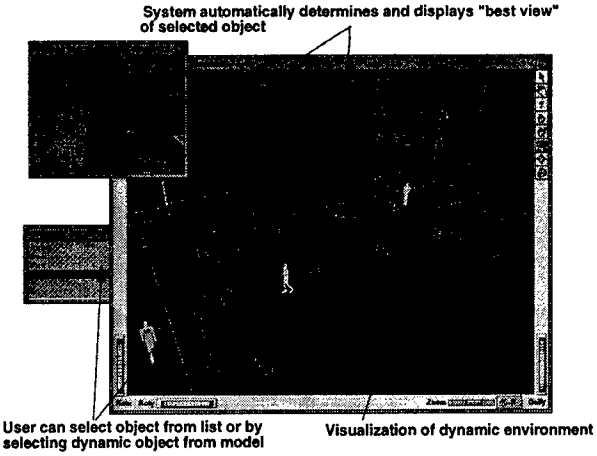


Figure 3: Current MPI Video interface, showing a graphical model of the environment, video stream (one of 6 total cameras is shown) and the list of objects currently in the environment.

3. A query environment which supports navigation and querying of the wealth of data input to and derived by the system[Tai, 1996, Santini and Jain, 1996].

Clearly, all three components of this system are equally important. In this paper, our focus is only on the first aspect above. That is, innovative techniques for video data analysis, including activity understanding and object recognition, and assimilation of the multiple streams of the data into the Environment Model (EM), segmentation and parameter extraction of articulated objects for activity recognition, and visualization. This is a crucial component of our *MPI-Video* system. The EM is coherent, dynamic, multi-layered, three-dimensional representation of the content in the video streams. It is this view-independent, task-dependent model that bridges the gap between two-dimensional image arrays which by themselves have little meaning and the complex information requirements placed by users and other components on the system.

Additional information, including implementation details, is available in technical reports [Chatterjee *et al.*, 1994, Katkere *et al.*, 1995].

5 Current Research

We are addressing many research issues in image understanding relevant to the VSAM to exploit the MPI video infra-structure already developed. These research issues will allow extension of the MPI video paradigm to satisfy needs of VSAM and will also strengthen MPI technology for other applications. In particular, our focus is the formalization of the Environment Model and its attendant vision processing algorithms.

5.1 Assimilation of Motion and Optical Flow

As the size and speed of computers increase, the use of motion is playing an increasingly important role in computer vision systems. Motion will play a central role in visual surveillance. In this section we describe our approach for the assimilation of motion into our surveillance system.

5.1.1 Generalized Shape-of-Motion Features

The significance of motion in human perception is highlighted throughout the psychophysical literature. Examples include Johansson's work on moving light displays (MLD) and more recently, Bertenthal and Pinto's [1993] work on the perception of the human gait. Cedras and Shah [1995] survey recent results regarding motion in computer vision systems. A defining characteristic of a motion interpretation system is whether or not it is *model-based.* While all vision systems use models of some sort to perform a task, in this context we refer to kinematic models of the moving object. For example, Ju, Black and Yacoob [1996, 1997] use a model-based approach that couples optical flow with an articulated model of a human. Hunter, Kelly and Jain [1997] use mixture models that are constrained to represent a collection of limbs and the torso of the human body. Polana and Nelson [1994, 1995], Baumberg and Hogg [1993, 1995], Little and Boyd [1995, 1996], and Bobick and Davis [1996a, 1996b] describe a model-free approaches to motion recognition.

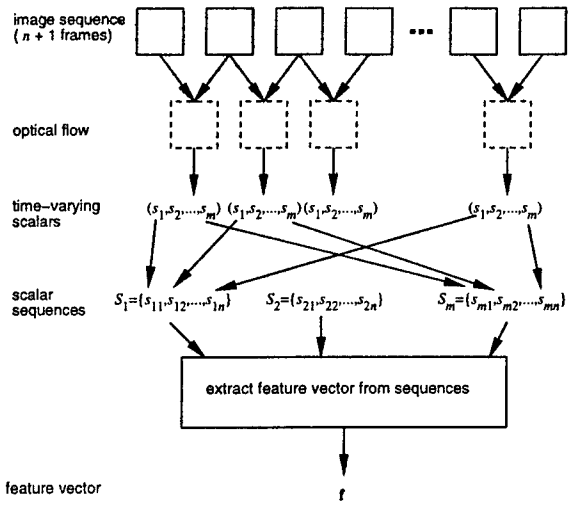


Figure 4: Generalized shape-of-motion feature extraction. An image sequence is reduced to a feature vector, **f**, based on the sequences of scalars that characterize the distribution of optical flow.

Our system begins with low-level model-free interpretation and progressively adds more information as is appropriate.

Figure 4 shows a generalized version of the system Little and Boyd [1995, 1996] use to create shape-of-motion features from a sequence of images that forms the basis of our motion recognition. The system begins with a sequence of images and derives dense optical flow. For each of the flow images, the system computes characteristics that describe the shape of motion (i.e., the spatial distribution of the flow). Rearranging the scalar values forms a time series for each scalar. The time series are converted into a feature vector used for recognition. For gait recognition, the conversion results in features that are the relative phases of the oscillating scalar values.

A more general approach looks for other patterns in the optical flow. Here we are inspired by the work of Johansson [1975] and Nogawa, Nakajima, Sato and Tamura [1997]. They look for patterns of optical flow related to specific motion stimuli. For example, Johansson showed that the outline of a shrinking box has several possible perceptions. Two possible perceptions are a box being compressed, or a box rotating in space. The shrinking box has a corresponding optical flow pattern, a region of flow that converges towards a central point and shrinking in size. Detecting such a pattern of optical flow in a subsequent sequence indicates the set of possible perceptions.

Many examples of flow patterns and perceptions exist in psychophysical and computer vision literature and more can be created. We plan to build a catalogue of such flow patterns. Then, using the system described in Figure 4, we will be able to construct a rich feature vector based on a wide variety of flow patterns. This enhanced feature vector will allow the system to interpret a broad range of motion, not just human gaits.

5.1.2 Motion Assimilation from Multiple Viewpoints

Our surveillance system will have multiple cameras available to view motion in a scene. We intend to exploit that abundance of data using an Environment Model (EM) assimilation system, as described in the following.

Katkere and Jain [1996] describe the *environment model* (EM) paradigm illustrated in Figure 5. The EM represents the state of the world, and the assimilation system iteratively updates the representation. An arbitrary number of sensors acquire information about the world in the form of measurement data. At specific moments in time, assimilator modules take the measurement data and incorporate it into the environment model (Figure 5(a)) by extrapolating the current state of the EM and updating the extrapolation to reflect the new data (Figure 5(b)).

The Kalman filter is closely related to the EM paradigm. The extrapolate-and-update data flow shown in Figure 5 is common to both, and we use the Kalman filter as a mathematical foundation for EM assimilation. In is usual form, the Kalman filter estimates a single time-varying state variable and a single source of measurements [Gelb, 1974]. In contrast, an *assimilating Kalman filter* allows multiple states

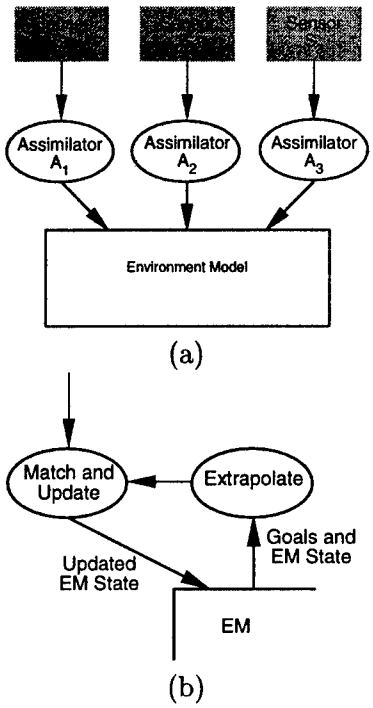


Figure 5: Illustration of environment model paradigm for information assimilation: (a) the overview of the system, and (b) the assimilator modules.

and data sources. Let $X_k = \{\mathbf{x}_{k,i}\}$ and $Z_k = \{\mathbf{z}_{k,j}\}$ be the sets of state and measurement vectors respectively for time k. The extrapolation stage is nearly identical to that for the simple Kalman filter, i.e.,

$$\hat{\mathbf{x}}^-_{k,i} = \Phi_{k-1,i}\hat{\mathbf{x}}_{k-1,i} + \mathbf{w}_{k-1,i}, \text{ and} \quad (1)$$

$$P^-_{k,i} = \Phi_{k-1,i}P_{k-1,i}\Phi_{k-1,i} + Q_{k-1,i}, \quad (2)$$

for all $i = 1, 2, \ldots, |X_k|$, where Φ is the system model, P is the covariance of the state, $\mathbf{w}$ and Q are the extrapolation error then the associated covariance. The superscript "$-$" indicates an extrapolated value before update. The update stage requires matching of incoming data with an individual state variable. Let $M_{k,i}$ be the set of indices for measurements that match state vector i at time k. The update is performed for all matches, i.e., for all $i = 1, 2, \ldots, |X_k|$, for all $j \in M_{k,i}$

$$\hat{\mathbf{x}}_{k,i} = \hat{\mathbf{x}}^-_{k,i} + K_{k,i,j}[\mathbf{z}_{k,j} - H_{k,j}\hat{\mathbf{x}}^-_{k,i}], \quad (3)$$

$$P_{k,i} = [I - K_{k,i,j}H_{k,j}]P^-_{k,i}, \text{ and} \quad (4)$$

$$K_{k,i,j} = P^-_{k,i}H^T_{k,j}[H_{k,j}P^-_{k,i}H^T_{k,j} + R_{k,j}]^{-1}, \quad (5)$$

where H is the measurement model, R is the measurement error covariance, and K is the Kalman gain matrix. If there are no matches for a particular state vector, then the update is trivial. Ayache and Faugeras [1988], and Matthies, Kanade and Szeliski [1989] give examples of data fusion systems that use a Kalman filter approach, like that described by Equations 1 through 5 to assimilate data.

Note that the assimilating Kalman filter is strongly coupled [Clark and Yuille, 1990]. Coupling of the independent measurements is innate to the Kalman filter because it does not care if different measurements are from the same sensor at different times or the different sensors at the same time. The filter allows high-level information to be incorporated into the system as additional sources of data.

We introduce a level of abstraction to the *assimilating Kalman filter* to create a *symbolic Kalman filter*. The state variable becomes feature vectors and perceptions, while the covariance matrix becomes the associated confidence measures. The resulting system is no longer strictly a linear system, but it does maintain the feature of strong coupling.

In our assimilation of motion, the state variable becomes

$$\hat{\mathbf{x}}_{k,i} = \left[\begin{array}{ccc} \mathbf{f}_{1,k,i} & \mathbf{f}_{2,k,i} & \mathbf{p}_{k,i} \end{array} \right]^T, \quad (6)$$

where $\mathbf{f}_{l,k,i}$ is the shape-of-motion feature vector for object i at time k as viewed by camera l. $\mathbf{p}_{k,i}$ is a perception vector, $p_{m,k,i} \in \{0, 1\}$ and indicates whether perception m is true or false. There is no covariance matrix as such. However, there is variance associated with each element of the feature vectors and a confidence level in the associated perception.

We can now illustrate the operation of the proposed motion assimilation system with a hypothetical example based on Johansson's shrinking box stimulus. The system assimilates motion from two cameras as follows.

1. Camera #1 views the scene from the front and produces the feature vector, $\mathbf{f}_{1,k,i}$. If

there are three valid perceptions, then the assimilation module produces the following state vector

$$\hat{\mathbf{x}}_{k,1} = \left[\begin{array}{ccccc} \mathbf{f}_{1,k,1} & \mathbf{0} & 1 & 1 & 1 \end{array} \right]^T.$$

2. Camera #2 views the scene from the side and produces a new value for $\mathbf{f}_{2,k,1}$. Suppose the features indicate that there is a counter-clockwise rotation for the object so only one of the three perceptions is possible. The state vector is updated to

$$\hat{\mathbf{x}}_{k,1} = \left[\begin{array}{ccccc} \mathbf{f}_{1,k,1} & \mathbf{f}_{2,k,1} & 0 & 1 & 0 \end{array} \right]^T.$$

   The system has a set of features and a unique perception of the motion in the scene.

3. In a larger system, additional assimilators could modify the perceptions contained in the EM as dictated by the feature vectors that they acquire. The extrapolation step in this system is simply an identity operation. In a more complicated and dynamic environment, extrapolation may be more complicated.

Using the EM paradigm to assimilate the data allows us to add data from disparate sources too. Any sensor that can indicate activity in a scene can be added to the system and contribute a feature vector and a set of perceptions.

5.1.3 High-Level Information

The low-level interpretation based on shape-of-motion features is limited in its use for reasoning about the scene. The features are adequate to reason about some things, but will not help answer questions that require, for example, structural information about a human. We contend that use of models in interpreting a scene depends entirely on what you need to know about a scene. Complex models need not be introduced until the information they provide is needed. Furthermore, before such a model can be applied, it must be known that the model is appropriate. An advantage of the EM paradigm is that we can introduce such models as the needs of the system dictate.

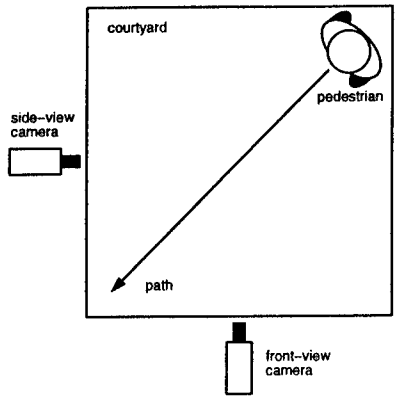


Figure 6: Plan view of a courtyard scene with a pedestrian walking along a diagonal path. The scene is observed by two cameras.

The following example illustrates how the proposed system will employ high-level *a priori* knowledge. Suppose that we are monitoring a courtyard with two cameras, Figure 6, both at eye level and aimed in the horizontal plane. A person walks diagonally across the courtyard. The EM assimilation system interprets the scene in the following steps.

1. The assimilator module for the front view camera builds a shape-of-motion feature vector and determines that there is an object, probably a person, moving from right to left across the courtyard and sets the perception vector accordingly.

2. The assimilator for the side camera also identifies the moving object, but sees it as moving from left to right. The module, based on the new information about the motion, updates the EM to indicate that a person is walking diagonally across the courtyard, since diagonal motion is the only thing consistent with the information already in the EM and the motion observed by the second camera.

3. Suppose we desire to know if pedestrians crossing the courtyard are carrying briefcases. To answer this question we need to know where the hands are, and for that we need the structure of the body. A high-level assimilator module is introduced to

determine kinematic structure. This module identifies the limbs and body of the person and adds another set of features, $\mathbf{f}$, to the state vector, $\hat{\mathbf{x}}$. These new features are the structural parameters of the body. From the kinematic structure the system locates the hands and determines whether or not a briefcase is present.

The system only introduces the model in step 3 when it is appropriate and needs the information that the model can provide.

5.2 Kinematic Structure from Constrained Mixture Models

Recovery of three-dimensional time-varying posture of complex articulated objects from two-dimensional image sequences is a basic requirement for the development of many important image sequence understanding applications. Examples include motion recognition systems, content-based addressing of video databases, and advanced human-computer interfaces. We are developing a novel articulated object posture and motion estimation framework that unites low-level uncertainty management techniques with explicit representation and enforcement of known object structure. *Articulated motion* is motion that arises from the movements of an articulated object assembled in joint-link fashion where the component bodies (links) are assumed to be rigid objects of fixed proportions. Currently, the observer is assumed stationary.

Low level processing consists of Bayesian maximum a posteriori (MAP) foreground/background segmentation followed by soft-labeling (probabilistic classification) of foreground observations to object structural components, which are modeled as a mixture density in the image space. Object kinematic structure is expressed as a set of constraint equations over the mixture parameter space that all valid postures must satisfy. Operationally, maximum likelihood estimation is achieved by employing a modified Expectation Maximization (EM) algorithm, called the Expectation-Constrained Maximization (ECM) algorithm, that projects every EM iterate into the feasible posture space.

The key concept in mixture density modeling of articulated object motion is the association of observation processes with object components. For example, an articulated object will have processes associated with each rigid link, which are taken to be responsible for the production of segmented foreground pixel data (or other basic observables). Once we have made this association, articulated object kinematic structure may be expressed as constraints over the space of mixture density parameters. This approach unites uncertainty management in early processing (via stochastic observation processes) with explicit, deterministic knowledge of object structure in an attempt to address the following design objectives:

1. Explicit representation and use of object kinematic models, and explicit 3D posture estimates.
2. Distinction between model acquisition and model-based estimation.
3. Model closely coupled with early vision (segmentation) via top down processing.
4. Formal uncertainty management at pixel level accounting for segmentation and component-wise pixel labeling errors, component shape variability and local deformations.
5. Natural and robust occlusion reasoning framework.
6. Explicit modeling and robustness to correlated segmentation dropout.
7. Equal applicability to single or multiple observer datasets.
8. Applicable to uncontrived environments and objects.
9. Extensible to use of object dynamics models and higher-level motion analysis models.
10. Applicable for guidance or control of finer segmentation techniques (e.g. snakes).

We believe these objectives represent the essential aspects of articulated motion estimation with regard to most applications.

We have begun to evaluate our computational framework on real data of the motion of a human being in an arbitrary indoor environment. Early tests show the algorithm returns subjectively good posture estimates in this uncontrived test environment, and that those estimates are always valid object postures.

5.2.1 Object Models, Postures and Orthographic Observers

For an articulated object with n rigid links (components) and m joints, and assuming the three-dimensional (unobservable) component observations have the Gaussian form, the articulated structural model is specified by the pair

$$\mathcal{S} = \{\Lambda, \mathsf{E}\} \tag{7}$$

where $\Lambda = \{\Lambda_i\}_{i=1,2,...,n}$ is the labeled set of diagonalized covariance matrices $\Lambda_i = \text{diag}(\sigma_{i,1}^2, \sigma_{i,2}^2, \sigma_{i,3}^2)$ (descending), one associated with each rigid link to characterize it's fixed three-dimensional shape; $\mathsf{E} = \{(j, k, \mathbf{a}_{j,k}, \mathbf{a}_{k,j})_l\}_{l=1,2,...,m}$ with $j, k \in \{1, 2, ..., n\}$, $j \neq k$ and $\mathbf{a}_{j,k}, \mathbf{a}_{k,j} \in \mathbf{R}^3$ denotes the connectivity structure of the object's m joints: component j is connected to component k, where the joint is located at $\mathbf{a}_{j,k}$ with respect to component j's reference frame, and $\mathbf{a}_{k,j}$ with respect to component k's reference frame.

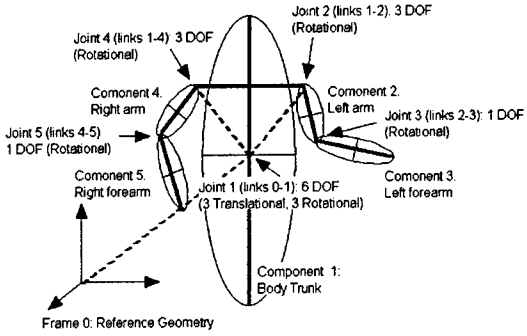
The articulated posture is defined to be the set

$$\mathbf{\Omega} = \{\{\theta_{i,1}, \theta_{i,2}, \theta_{i,3}\}_{i=1,...,n}, \mu_{1,1}, \mu_{1,2}, \mu_{1,3}\} \tag{8}$$

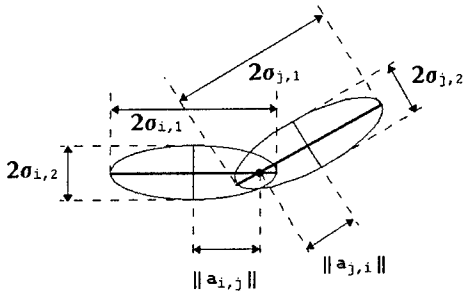
of Euler joint angles for each joint, and the position μ_1 of the base component. The posture Ω implies a set

$$\mathbf{\Phi} = \{\phi_i\}_{i=1,2,...,m} \tag{9}$$

of labeled mixture process parameters associated with individual rigid components comprising the articulated body. Here, $\phi_i = \{\mu_i, \Sigma_i\}$ with mean vector $\mu_i \in \mathrm{R}^3$ and covariance matrix $\Sigma_i \in \mathrm{R}^{3\times3}$, related to the eigenvalue matrix



(a)



(b)

Figure 7: (a) 14-degree-of-freedom structural model for analysis of image sequences containing human beings. (b) Schematic diagram of the mixture process associated with a two component, single joint articulated object (e.g. a potion of the object in (a)).

by $\Sigma_i = R_{0,i}\Lambda_i R_{0,i}^T$ for rotation $R_{0,i}(\mathbf{\Omega})$ with respect to the reference frame.

The set $\mathbf{\Omega}$ is a valid posture according to model $\mathcal{S}$ if each component has the correct eigenvalues Λ_i, and the collection of components satisfy the joint conditions in $\mathcal{S}$. Posture is not generally observable because available image data is two-dimensional. Images may be interpreted as arising from a *marginalized* data source, with the dimension corresponding to the optical axis integrated out, assuming orthographic projection in image formation. These 2D observation processes are responsible for the generation of pixel foreground observations (coordinate pairs of foreground pixels) at the output of a segmentation algorithm.

Assuming orthographic image projection and, without loss of generality, taking the $x_1 - x_2$ plane as the imaging plane and x_3 as the optical axis, the observation process corresponding to component i is the two-dimensional (spatial) Gaussian density function with parameters $\phi_i' = \{\mu_i', \Sigma_i'\}$, μ_i' = upper 2-vector of μ_i, Σ_i' = upper left 2×2 sub-matrix of $\Sigma_i = R_{0,i}\Lambda_i R_{0,i}^T$. Assuming observations of component i are drawn according to the observation process that distributes them spatially according to it's parameters ϕ_i', then arbitrary (i.e. unlabeled) data can be taken to arise from the mixture process

$$p(w_i|\Phi') = \sum_{j=1}^{n} \alpha_j p_j(w_i|\phi_j') + \alpha_{\mathrm{BG}} \qquad (10)$$

where α_j is the prior probability of observing component j at an arbitrary image site, and α_{BG} is a uniform outlier process, $\sum_{j=1}^{n} \alpha_j + \alpha_{\mathrm{BG}} = 1$ such that all object component processes have equivalent peak values.

5.2.2 Integrating Known Kinematic Structure into Maximum-Likelihood Posture Estimation: ECM Algorithms

The EM algorithm does not, in it's common form, account for explicit parameter space dependencies amongst elements of Φ[1]. One may account for this, however, by viewing the maximization step of the EM algorithm as a *constrained* optimization procedure, given the a priori structural knowledge in the model $\mathcal{S}$. We refer to this modified EM procedure as an *Expectation-Constrained Maximization* (ECM) algorithm.

A full set of constraint equations, $\mathbf{c} = \mathbf{0}$ describing each joint with respect to it's constituent components, define a nonlinear subspace (the "feasible" space or constraint manifold) in the total parameter search space. The correct expression for joints between components i and j is

$$\mu_i + \mathbf{d}_{i,j} r_{i,j} = \mu_j + \mathbf{d}_{j,i} r_{j,i} \qquad (11)$$

where $\mathbf{d}_{i,j} r_{i,j}$ is the orientation of component i's joint axis with respect to the reference frame, $r_{i,j}$ is the distance from the component mean to the joint along $\mathbf{d}_{i,j} r_{i,j}$, etc. Equation 11 results in three constraint relations per joint. To enforce these constraints, each EM iterate is projected onto this subspace by a Newton-Raphson (NR) procedure [Fletcher, 1987]. This is equivalent to a generalized elimination of variables and effects a search exclusively in the $\mathbf{R}^{d-s}$ feasible subspace (for d parameters and s constraints). By requiring every step to remain in this subspace, we are guaranteed that the constraints, and thus our a priori knowledge and structural model $\mathcal{S}$, will not be violated.

The projection procedure, sketched geometrically in Figure 8, is repeated until a suitable termination criteria is met, such as negligible projected step size, which is the least squares optimal constrained solution.

Figure 9 shows input, segmentation and posture estimate images for 5 frames of a test sequence using a 5-component human being model as in Figure 7. We are currently extending our algorithms to conduct an extensive experimental analysis of the posture estimation model proposed here.

[1] Although, as indicated in [Redner and Walker, 1984], convergence results do hold for families of probability densities whose parameters are not independent of one another.

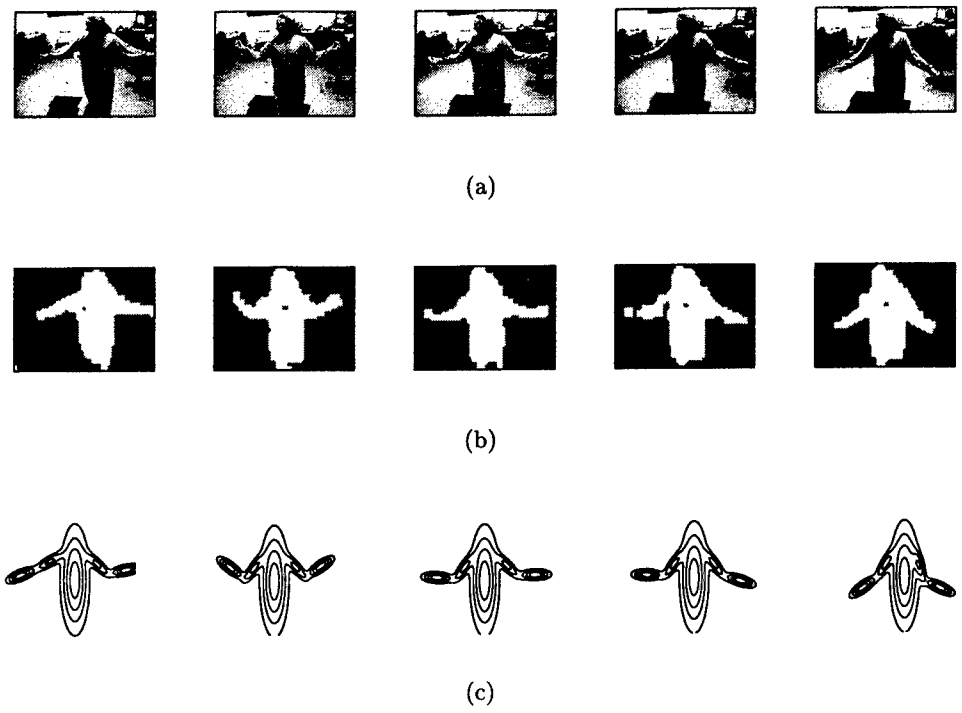


Figure 9: The input images in this test sequence (a) were analyzed using a 5-component 14 DOF articulated human model to recover the result MAP segmentation (b), and articulated motion (time-varying postures) of the object (c). The graphs in (c) are contours of the estimated mixture process, with components fit to rigid object links (2 forearms, 2 arms, and body trunk).

5.3 Event Recognition Using HMM

The objective of motion understanding is to provide computers with the ability to identify motions captured by video cameras. The definition of what is a motion varies across applications. For example, in a dance application the motions could be the possible movements (plié, pirouette, etc.) in ballet. A characteristic of motions that does extend across applications is the fact that they occur in the spatio-temporal domain where both the evolving shape and the trajectory of the object can be expressed. Motion understanding differs from the classical problem of object tracking in that for tracking, the output of the system is a trajectory defining the position of the object in time. The output of a motion understanding system is a sequence of semantic labels (typically verbs) describing the motions identified in the video sequence. Motivating the development of motion understanding is the possibility of applications such as computerized sports analysis, *immersive video*, intelligent surveillance systems and intuitive machine interfaces.

Our approach to motion understanding exploits finite state estimation. The new algorithm is a hidden Markov model, probabilistic, feature-based technique whose purpose is to identify the motions performed in an image sequence [Schlenzig", 1997].

The assumptions which make this technique feasible are:

1. **The motions we are interested in identifying can be enumerated.** The system is armed with a set of possible motions from which it must choose the best one. For the case where the image sequence contains a completely novel motion, the system returns an *unknown* response. On

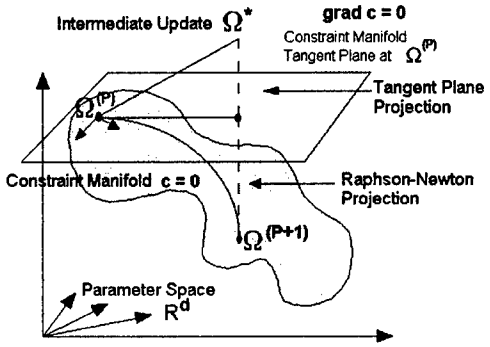


Figure 8: Geometry of the ECM algorithm. At each iteration, the ECM must produce a valid posture estimate by projecting the intermediate EM estimate $\Omega^\star$ onto the feasible space $\mathbf{c} = \mathbf{0}$. This is implemented as a two-step procedure at each iteration: (1) jump in the constraint manifold tangent plane in the direction of the EM iterate projection, followed by (2) projection in the tangent plane complement space back onto the constraint manifold.

line learning of new motions is not possible, but the architecture of the system allows new motions to be incorporated without the need to retrain on all possible motions. Instead only the transition probabilities for the new motion must be determined.

2. **Each motion can be described as a time sequence of static poses.** The location of the centroid of the object in motion can also be considered a "pose". This allows us to recognize activities of a rigid body undergoing translation. Some applications will require that both the shape of the object and its location be used in determining the observation symbol.

3. **A technique exists for distinguishing amongst the poses.** Image processing must provide robust determination of the shape of the object and/or the location of its centroid. For the system to be useful, recognition must occur across a significant variety of instantiations.

No assumptions are made about the sequence containing a single motion. Instead, we depend on the algorithm to cope with and identify motion transitions.

The motion estimator, $\hat{\phi}_n$ is assumed to have the form

$$\hat{\phi}_n = E\{\phi_n | Z_n\} \tag{12}$$

where ϕ_n is an indicator vector describing the true gesture information and Z_n is the measurement data available at time n. The derivation of the estimator [Krishnan, 1984, Sworder, 1991, Sworder *et al.*, 1995] yields the update equation

$$\begin{aligned} d\hat{\phi}_n &= Q'\hat{\phi}_n\, dt + P_{\phi\phi}\lambda D' R_n^{-1}\, d\nu_n \quad (13) \\ P_{\phi\phi} &= \phi_n\hat{\phi}_n' - \hat{\phi}_n\hat{\phi}_n' \\ R_n &= diag(\lambda D\hat{\phi}_n') \end{aligned}$$

where the first right-hand term in Equation 13 is the effect of the model and the second term provides the change due to the current measurement. In Equation 13, $d\nu$, is the vector e_i where i is the current pose symbol, λ is the image frame capture rate and $dt = 1$. The discernibility matrix, D, contains the probabilities describing the observation process.

The first step in using the finite state estimator for motion understanding involves determining the parameters of the hidden Markov model. Previous attempts at using HMMs for image sequence analysis have used techniques which iteratively determine the number of states, what the states are, and the transition probabilities based on training data. These methods, while easy to use, destroy some of the greatest benefits of using HMMs. Typically the results do not even allow one to clearly distinguish what has been defined to be a state. This prevents the designer from incorporating *a priori* known information into the estimator.

In designing the estimator, the user has two powerful sets of parameters with which to influence the behavior of the system. The diagonal elements of the rate matrix, Q, are proportional to the mean performance time for each activity. The off diagonal elements represent the probabilities of changing from a pose in motion A to a pose in motion B. Because the motions are dictated by the application and the poses are chosen by the designer, this data is easily obtained from sample sequences.

The discernibility matrix, D, encodes how well the image processor is able to distinguish amongst the expected poses. It also allows the designer to implicitly describe the classification accuracy and the resolution available in locating the centroid of the object. The discernibility matrix has a predictable effect on the estimator. If the designer is overly conservative in his expectations of the performance of the image processor, as evidenced by low probabilities of correct identification, then the system will respond by somewhat disregarding the measured information at each time step. This will slow down the transitions from one motion to the next. If the discernibility matrix is the identity matrix then each symbol observation will be assumed to be always correct, and if errors actually do occur, the estimator will typically incorrectly indicate a motion transition.

In addition to the selection of parameters, the designer must choose the method by which the system probabilities will be interpreted. The most straightforward approach is to simply respond with the activity associated with the maximum likelihood, but this has several shortcomings. First, the system will always respond with an answer from the set of possibilities. This contradicts our earlier statement that in the case of a novel activity occurring in the image sequence the system should respond with an *unknown*. To achieve this there must be a lower bound set on what is an acceptable level of confidence for an activity before it is identified as the chosen one. It also makes sense to require a minimum distance between the maximum likelihood and it's nearest neighbor. For example, if there exist three possible activities and the current probabilities of them occurring are $[0.5, 0.45, 0.05]$ respectively then we would not want the system to confidently proclaim that the first activity is occurring. Instead, the system should admit to confusion, and wait for additional measurements which will hopefully make things clearer. Access to the probability of each activity also allows us to impose a risk function on the decision making process. This enables us to incorporate *a priori* known information of the type "it's better to err by saying motion A is occurring if motion B is actually happening than it is to err in saying motion B is occurring when it isn't".

To illustrate the usage of the estimator, consider the following example that illustrates the use of the filter for the 3 motion/4 pose system given in Figure 10. Here the set of recognizable motions is $\theta_t = \{A, B, C\}$ and the set of poses is $\rho'_t = \{1, 2, 3, 4\}$. The Kronecker product of the two yields $\phi'_t = \{\ A1,\ A2,\ A3,\ A4,\ B1,\ B2,\ B3,\ B4,\ C1,\ C2,\ C3,\ C4\ \}$. Motion A consists of a sequence of pose 1 followed by pose 2 while motion B is defined to be a sequence of pose 2 followed by a sequence of pose 3, and motion C is defined to be a sequence of pose 1 followed by a sequence of pose 4.

The rate matrix, Q, is found by first identifying the expected execution time for each motion. For this example, it was decided that each pose is expected to last around 4.2 time steps. Therefore, the diagonal elements of the rate matrix are 0.24. To validate the performance of the Q matrix one can run the filter with no measurement inputs and verify that the motion proba-

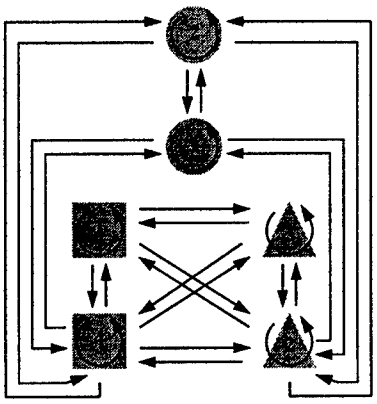


Figure 10: The first example consists of three types of motion (indicated by shape) and four possible poses (indicated by number). Motion A (circles) consists of pose 1 and pose 2, motion B (triangles) includes both pose 2 and pose 3, and motion C (squares) is made up of pose 1 and pose 4.

bilities achieve the expected equilibrium states.

The upper graph in Figure 11 gives the observation symbol at each time step using the given Q and D. Observation errors occur at time steps 50 and 54. The evolutions of the probabilities for each of the possible motions are given in Figure 12. The output of the filter is illustrated as a solid line in the lower graph in Figure 13. For this example, the maximum likelihood was used to identify the current motion. If the maximum likelihood was less than 0.5 then the output of the system was set to *unknown* (indicated by $*$ in the figure). The true motion is shown in the upper graph of figure 13.

The results show that the system is uncertain at times of motion transitions which causes some delay in identifying an event. This delay is insignificant when compared to the latency that exits in batch processing systems where the user would have to wait until the termination of a motion before analysis could begin.

Note that around time step 50 the filter makes an error. This is caused by two factors. First, there is a long sequence of observations of pose 1, a pose that is shared between motions A and C. Second, this is followed by an incorrect observation of pose 4. This causes a spike in the probability of motion C which is large enough to confuse the system.

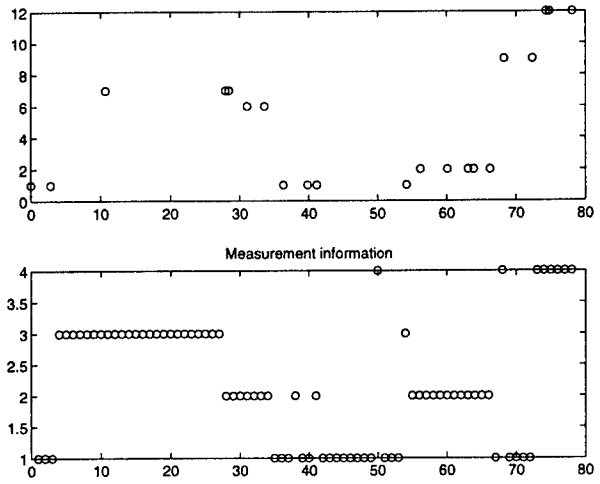


Figure 11: The input data generated using the given Q and D. The upper graph shows only the transitions from motion i to motion j, $i \neq j$. The lower graph shows the observations for all time steps.

Of particular importance in the application of finite state estimation to the problem of motion understanding is the ease in which a new collection of motions can be modeled and identified. Given a sampled sequence of images one can determine the mean execution times and the transition probabilities. Furthermore, the designer has the means to easily incorporate *a priori* known information into the system using both the rate transition matrix and the discernibility matrix. The designer can imbue additional information and constraints through the limiting of feasible transitions. For example, a motion grammar could be imposed upon the system such that motion B can only occur after motion A. With such flexibility it is expected that finite state estimation will permit real world applications of motion understanding.

5.4 Three-Dimensional Modeling and Visualization

Conventional videos present image sequences seen from predetermined camera viewpoints. With the advancement of image processing and

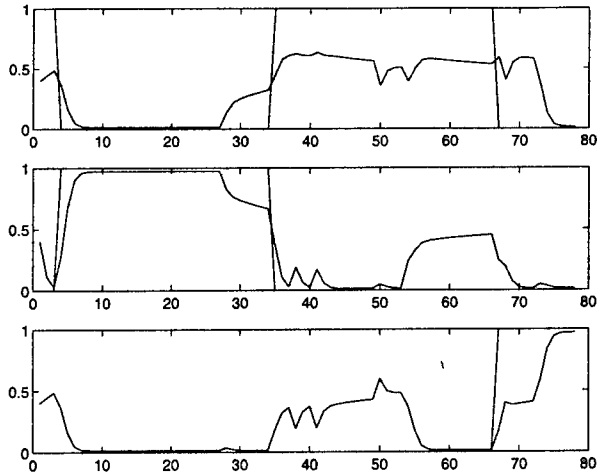


Figure 12: Each graph presents the evolution of the probability of a given motion. The actual motions are shown in red.

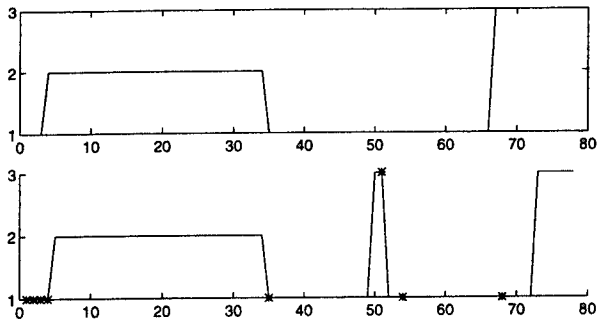


Figure 13: At each time step the probabilities of the motions are compared to select the most likely (maximum probability). If the maximum probability is greater than 0.5 the system confidently identifies the current motion. In the case where the maximum probability is not greater than 0.5, the system admits confusion.

generation techniques, it is now possible to capture spatial-temporal models of real scenes and events, and these recordings can be watched in a three-dimensional fashion, with the viewpoints controlled by the viewer during playback. 3D video possesses the interactive characteristics of virtual reality systems, yet its contents are based on real environments.

The key issue of 3D video generation is the capture and modeling of real scene geometry and the rendition of the views from arbitrary perspectives. Generally, researchers take two types of approaches toward the virtual view synthesis problem: model-based approaches, and image-domain methods. Model-based approaches first try to recover the 3D model of the environment, including object shapes and colors. Then for playback, the models are rendered from the desired viewpoint. Works in this category include [Kanade *et al.*, 1995, Tseng and Anastassiou, 1995, Fuchs *et al.*, 1994]. Image-domain methods avoid the issue of detailed shape recovery but instead generate the new views via direct image transformation on existing camera views. Works of this type include [Chen and Williams, 1993, Skerjanc and Liu, 1991, Seitz and Dyer, 1995, McMillan and Bishop, 1995, Levoy and Hanrahan, 1996, Gortler *et al.*, 1996]. Hybrid methods, combining the characteristics of each, also exist. One such work is [Debevec *et al.*, 1996].

In this research the virtual view generation methods are based on the *MPI-Video* framework. An important assumption in *MPI-Video* is that the knowledge about the static environment is known *a prior*, therefore the problem transforms to the segmentation and representation of dynamic objects. Another assumption, that cameras are stationary, greatly simplify the complexity of virtual view generation. With precise camera calibration to obtain their parameters expressed in a common world coordinate system, *MPI-Video* can support realistic 3D video, *immersive video*, using both modeling-based and image-domain approaches.

5.4.1 The Model-based Approach

Because the precise location of cameras is known, the extent of volume occupied by dynamic objects can be accurately determined. Our approach, shown in Figure 14, employs voxel occupancy determination to obtain full 3D models of dynamic objects, and an efficient method for "painting" the shape using real camera views. The result is a highly effective and realistic presentation of dynamic video with full 3D viewpoint control.

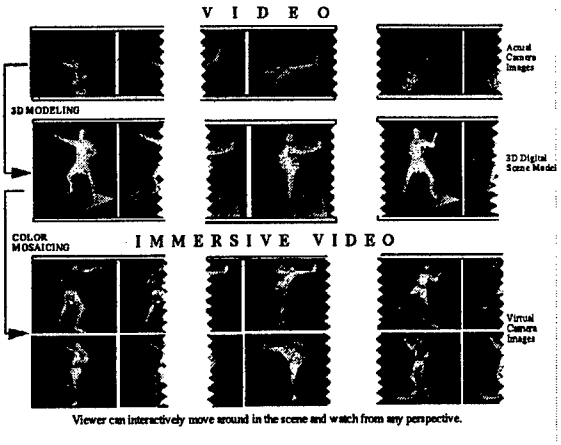


Figure 14: Overview of the model-based approach. Given multiple camera images of the same scene (top), the 3D shapes of the scene object are first reconstructed (middle), and then the surface colors are determined, resulting in a photo-realistic 3D video sequence (bottom)

The procedure to create model-based *immersive video* involves proper studio setup and data acquisition, camera calibration, object segmentation, shape recovery, color mosaicing, and playback. The last four steps, for model generation and rendering, are briefly explained below. For a detailed discussion, see [Moezzi *et al.*, 1997].

5.4.2 Object Segmentation

We can simply determine the extent of dynamic objects in each camera view by subtracting from them the known background images. Practically, noise in the data and shadow or reflections make segmentation much more difficult, so manual cleanup may be necessary. Development of more intelligent segmentation methods are part of the ongoing research in this project.

5.4.3 Shape Recovery

With the segmentation results from each camera, we can determine the dynamic object shape with volume intersection methods. Specifically, we divide the space into a set of voxels, and project rays from pixels in the "empty" region of each segmented camera images. As shown in Figure 15, all voxels hit by any such rays must be empty, and the remaining voxels represent the portion of space occupied by objects. To obtain efficient representation, we convert the voxel representation into a polygonal-surface-based one using the Marching Cube algorithm [Lorensen and Cline, 1987].

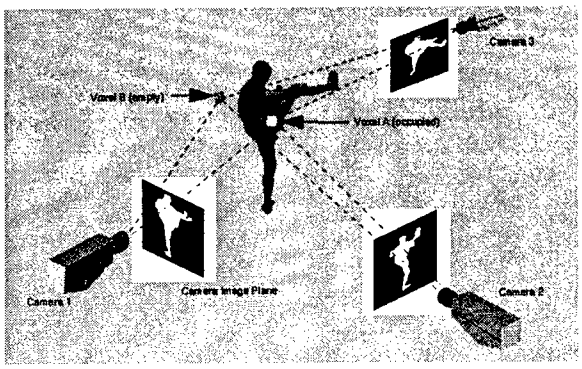


Figure 15: Voxel occupancy determination. Voxel A projects to "object-occupied" regions in all cameras and is determined occupied; Voxel B projects to "empty" regions in Cameras 1 and 2 and is declared empty.

5.4.4 Color Mosaicing

We have the object shape, but its surface colors are still unknown. We can simply project rays from each pixel in the object portions of real camera views into space, and the first polygon on the surface hit by the ray will acquire the pixel colors. But this method requires too much computation and is slow. Too speed up processing, we utilize "color mosaicing", or first coloring each polygon with a color value converted from its index. Then, taking advantage of the specialized graphic hardware in high-end workstations, we render the object model from the viewpoints of real cameras, and by comparing the renderings with real camera views pixel-by-pixel, we can easily determine the color of each polygon.

5.4.5 Playback

We determine the 3D object models in a frame-by-frame basis. Once we have models of all frames, we simply load them into memory and rapidly switch between them. Provided that the graphics hardware can render them rapidly, we can achieve real-time playback. A comparison between the resulting rendering and an original image is shown in Figure 16.

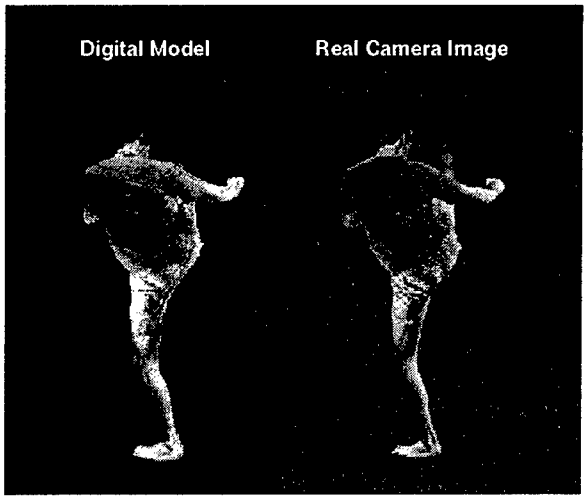


Figure 16: Comparsion between a real camera image (right) and the constructed 3D model rendered from the same viewpoint of the camera (left).

5.4.6 The Image-Domain Approach

While the model-based approach can produce excellent results for controlled environments, more general conditions require image-domain transformation due to limitation of segmentation techniques. Also, visibility for pixels need to be mapped correctly. If we choose cameras for which the monotonicity constraint [Seitz and Dyer, 1995] holds, it is guaranteed that physically correct results can be obtained. To ensure we can apply this constraint, we need to have a good approximation of the dynamic object shape.

5.4.7 The Interactive Modeler

Currently we are developing techniques that will utilize interactive manual fitting of simple blocks to shapes of dynamic objects. The approach taken is inspired by the works of Debevec, *et al.* [1996]. The interactive modeling program utilizes domain knowledge to guide the user during the modeling process.

5.4.8 Incremental Enhancement of Object Models

In video, neighboring frames often exhibit a high degree of coherence. By exploiting this rich source of information, we can check our rough model obtained previously and correct any "defects." By utilizing temporal knowledge, we can gradually refine the models and have accurate shape information.

5.4.9 Plenoptic Model-based Representation

With the approximations for object shape, the next task is to capture and generate view information. We are developing representations that are based on plenoptic functions [McMillan and Bishop, 1995, Levoy and Hanrahan, 1996, Gortler *et al.*, 1996]. By choosing the proper cameras we can re-construct the plenoptic field, and then virtual views are generated by sampling the proper subsets of the field.

6 Conclusions

We have described the infrastructure of *MPI-Video* in the context of VSAM application. *MPI-Video* relies on two important concepts. The first, content-based interactivity, allows a user of a system to access information based on context, thus allowing the user to retrieve useful information even when the volume of data available is large. The second, gestalt perception, refers the merging of data from multiple image sensors into a single perception that conveys more information than just the individual images. We have focused on the assimilation of multiple streams of video into a single, integrated representation of the world. This will form the basis for future work which includes building a database subsystem to support content-based query operations, and

a query environment which supports navigation and querying of the wealth of data input to, and derived by, the system.

Research issues we are pursuing to assimilate data for *MPI-Video* VSAM include assimilation of motion and optical flow, determination of kinematic structure of objects in a scene, event recognition, and three-dimensional visualization. Assimilation of motion and optical flow will allow *MPI-Video* to exploit the abundance of motion information available in multiple image sequences. In VSAM applications, understanding the activity of humans is important. Recovery of kinematic structure of humans and recognizing what they are doing facilitates context-based addressing of sequences involving human activity. Visualization methods permit easier user interaction with the *MPI-Video* VSAM system.

References

[Ayache and Faugeras, 1988] N. Ayache and O. D. Faugeras. Building, registrating, and fusing noisy visual maps. *The International Journal of Robotics Research*, 7(6):45–65, December 1988.

[Baumberg and Hogg, 1993] A. M. Baumberg and D. C. Hogg. Learning flexible models from image sequences. Technical Report 93.36, University of Leeds School of Computer Studies, October 1993.

[Baumberg and Hogg, 1995] A. M. Baumberg and D. C. Hogg. Learning spatiotemporal models from training examples. Technical Report 95.9, University of Leeds School of Computer Studies, March 1995.

[Bertenthal and Pinto, 1993] B. I. Bertenthal and J. Pinto. Complementary processes in the perception and production of human movements. In L. B. Smith and E. Thelen, editors, *A Dynamic Systems Approach to Development: Applications*, pages 209–239. MIT Press, Cambridge, MA, 1993.

[Black *et al.*, 1997] M. J. Black, S. X. Ju, and Y. Yacoob. Recognizing human motion using parameterized models of optical flow. to appear in Motion-Based Recognition, M. Shah and R. Jain eds., 1997.

[Bobick and Davis, 1996a] A. F. Bobick and J. W. Davis. An appearance-based representation of action. In *13th International Conference on Pattern Recognition*, Vienna, Austria, August 1996.

[Bobick and Davis, 1996b] A. F. Bobick and J. W. Davis. Real-time recognition of activity using temporal templates. In *Third International Workshop on Applications of Computer Vision*, Sarasota, Florida, December 1996.

[Cedras and Shah, 1995] C. Cedras and M. Shah. Motion-based recognition: a survey. *Image and Vision Computing*, 13(2):129–155, March 1995.

[Chatterjee *et al.*, 1994] Shankar Chatterjee, Ramesh Jain, Arun Katkere, Patrick Kelly, Don Y. Kuramura, and Saied Moezzi. Modeling and interactivity in *mpi-video*. Technical Report VCL-94-103, Visual Computing Laboratory, University of California, San Diego, December 1994.

[Chen and Williams, 1993] S. E. Chen and L. Williams. View interpolation for image synthesis. In *Proceedings of SIGGRAPH 93*, Los Angeles, CA, USA, August 1993.

[Clark and Yuille, 1990] J. J. Clark and A. L. Yuille. *Data Fusion for Sensory Information Processing Systems.* Kluwer Academic Publishers, Boston, 1990.

[Debevec *et al.*, 1996] Paul Debevec, Camillo. J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proceedings of SIGGRAPH 96*, New Orleans, Louisiana, USA, August 1996.

[Fletcher, 1987] R. Fletcher. *Practical Methods of Optimization.* John Wiley and Sons, 1987.

[Fuchs *et al.*, 1994] Henry Fuchs, Gary Bishop, Kevin Arthur, Leonard McMillan, Ruzena

Bajcsy, Sang Lee, Hany Farid, and Takeo Kanade. Virtual space teleconferencing using a sea of cameras. In *Proceedings of the First International Symposium on Medical Robotics and Computer Assisted Surgery*, pages 22–24, Pittsburgh, PA, USA, September 1994.

[Gelb, 1974] A. Gelb, editor. *Applied Optimal Estimation.* MIT Press, Cambridge, Massachusetts, 1974.

[Gortler *et al.*, 1996] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The Lumigraph. In *Proceedings of SIGGRAPH 96*, New Orleans, Louisiana, USA, 1996.

[Hunter *et al.*, 1997] E. Hunter, P. H. Kelly, and R. Jain. A unified mixture density framework for segmentation and estimation of articulated object posture in image sequences. submitted to IEEE Nonrigid and Articulated Motion Workshop, CVPR 97, January 1997.

[Jain and Hampapur, 1994] Ramesh Jain and Arun Hampapur. Metadata in video databases. In *Sigmod Record: Special Issue On Metadata For Digital Media.* December 1994.

[Jain and Wakimoto, 1995] Ramesh Jain and Koji Wakimoto. Multiple Perspective Interactive Video. In *Proceedings of International Conference on Multimedia Computing and System*, Washington, D. C., USA, May 1995.

[Johansson, 1975] G. Johansson. Visual motion perception. *Scientific American*, pages 76–88, June 1975.

[Ju *et al.*, 1996] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: a parameterized model of atriculated image motion. In *2nd International Conference on Automatic Face and Gesture Recognition*, pages 38–44, Killington, Vermont, October 1996.

[Kanade *et al.*, 1995] Takeo Kanade, P. J. Narayanan, and Peter W. Rander. Virtual*ized* reality: Concepts and early results. In *IEEE Workshop on the Representation of Visual Scenes*, Boston, MA, June24 1995.

[Katkere and Jain, 1996] A. Katkere and R. Jain. A framework for information assimilation. In M. S. Landy, L. T. Malorey, and M. Pavel, editors, *Exploratory Vision: The Active Eye.* Springer-Verlag, New York, 1996.

[Katkere *et al.*, 1995] Arun Katkere, Saied Moezzi, and Ramesh Jain. Global multi-perspective perception for autonomous mobile robots. In *Workshop for Vision for Robots, IROS '95*, Pittsburg, PA, August 1995. IEEE.

[Katkere *et al.*, 1997] Arun Katkere, Saied Moezzi, Don Kuramura, Patrick Kelly, and Ramesh Jain. Towards video-based immersive environments. In *ACM-Springer Multimedia Systems Journal: Special Issue on Multimedia and Multisensory Virtual Worlds.* Spring 1997.

[Kelly *et al.*, 1995] Patrick Kelly, Arun Katkere, Don Kuramura, Saied Moezzi, and Shankar Chatterjee. An architecture for multiple perspective interactive video. In *Proceedings of ACM Multimedia 95*, 1995.

[Krishnan, 1984] V. Krishnan. *Nonlinear Filtering and Smoothing: An Introduction to Martingales, Stochastic Integrals and Estimation.* John Wiley and Sons, New York, NY, 1984.

[Levoy and Hanrahan, 1996] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of SIGGRAPH 96*, New Orleans, LA, USA, August 1996.

[Little and Boyd, 1995] J. J. Little and J. E. Boyd. Describing motion for recognition. In *IEEE Symposium on Computer Vision*, pages 235–240, November 1995.

[Little and Boyd, 1996] J. J. Little and J. E. Boyd. Recognizing people by their gait: the shape of motion. submitted to Videre, December 1996.

[Lorensen and Cline, 1987] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, July 1987.

[Matthies *et al.*, 1989] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.

[McMillan and Bishop, 1995] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *Proceedings of SIGGRAPH 95*, Los Angeles, CA, USA, August 1995.

[Moezzi *et al.*, 1996] Saied Moezzi, Arun Katkere, Don Kuramura, and Ramesh Jain. Immersive video. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, Santa Clara, California, USA, March 1996.

[Moezzi *et al.*, 1997] Saied Moezzi, Li-Cheng Tai, and Philippe Gerard. Virtual view generation for 3d digital video. *IEEE Multimedia*, spring 1997.

[Nogawa *et al.*, 1997] H. Nogawa, Y. Nakajima, Y. Sato, and S. Tamura. Acquisition of symbolic description from flow fields: a new approach based on a fluid model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):58–63, January 1997.

[Polana and Nelson, 1994] R. Polana and R. Nelson. Recognition of nonrigid motion. In *1994 DARPA Image Understanding Workshop*, pages 1219–1224, 1994.

[Polana and Nelson, 1995] R. Polana and R. Nelson. Nonparametric recognition of nonrigid motion. Technical report, University of Rochester, 1995.

[Redner and Walker, 1984] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, 26(2):195–239, 1984.

[Santini and Jain, 1996] Simone Santini and Ramesh Jain. The graphical specification of similarity queries. *Journal of Visual Languaages and Computing*, 7:403–421, 1996.

[Schlenzig", 1997] "Jennifer Schlenzig". *"The Semantic Analysis of Motion by Nonlinear Estimation Methods"*. PhD thesis, "University of California, San Diego", "1997".

[Seitz and Dyer, 1995] Steven M. Seitz and Charles R. Dyer. Physically-valid view synthesis by image interpolation. In *Proceedings of Workshop on Representations of Visual Scenes*, Cambridge, MA, USA, 1995.

[Skerjanc and Liu, 1991] R. Skerjanc and J. Liu. A three camera approach for calculating disparity and synthesizing intermediate pictures. In *Signal Processing: Image Communication*, volume 4, pages 55–64, 1991.

[Swanberg *et al.*, 1993] Deborah Swanberg, C. F. Shu, and Ramesh Jain. Knowledge guided parsing in video sequences. In *Proceedings of SPIE: Storage and Retrieval for Image and Video Databases*, volume 1908, pages 13–25, San Jose, CA, USA, 1993.

[Sworder *et al.*, 1995] David D. Sworder, Mark Kent, Robert Vojak, and R. G. Hutchins. Renewal models for maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, pages 138–149, January 1995.

[Sworder, 1991] David D. Sworder. Tactical decision making under stress. Technical report, Naval Ocean Systems Center, 1991.

[Tai, 1996] Li-Cheng Tai. Hypermedia in Multiple Perspective Interactive Video. Visual Computing Laboratory, UCSD, 1996.

[Tseng and Anastassiou, 1995] B. L. Tseng and D. Anastassiou. A theoretical study on an accurate reconstruction of multiview images based on the viterbi algorithm. In *IEEE Int'l Conf. on Image Processing ICIP '95*, Washington, D.C., oct 1995.

Sketch-First Modeling of Buildings from Video Imagery

Bob Bolles, Marty Fischler, Marsha Jo Hannah, Tuan Luong*
Artificial Intelligence Center, SRI International
333 Ravenswood Ave., Menlo Park, CA 94025
Telephone: 415-859-4620, E-MAIL: bolles@ai.sri.com

Riadh Munjy, Mushtaq Hussain
Calgis, Inc., 1477 E. Shaw Ave., Ste. 110, Fresno, CA 93710
Telephone: 209-298-1816, E-MAIL: munjy@engr.csufresno.edu

Abstract

A technique for constructing detailed 3-D models of buildings from video and other data sources is described. In addition, a ground-truth model of a moderately complex building is presented, in conjunction with a description of a set of video sequences and still images that can be used for evaluating existing techniques or developing new ones. The proposed modeling technique is called sketch-first modeling because a user first constructs a quick sketch of the building to be modeled, and then an enhanced bundle-adjustment procedure computes the locations of the cameras and the 3-D locations of the sketched primitives. The sketch can be as weak as a set of building components and constraints on them (such as "vertical walls" or "perpendicular lines") or as strong as an approximate 3-D model. There are several advantages of starting with a sketch. First, it provides a way to directly incorporate known world constraints into the solution procedure. Second, it can help keep track of the portions of the building that have been imaged. Third, it can help identify matching mistakes by locating points that are not consistent with perspective imaging of planar faces. And fourth, it can help identify points that are not on the building, such as points on vegetation.

*This work was sponsored by the Office of Research and Development under ORD Contract No. 93-F151700-000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Government or SRI International.

1 Introduction

The computer graphics community has made dramatic progress in the generation of realistic images from three-dimensional (3-D) models. The new techniques, however, are not practical for application to such key military tasks as simulation and mission rehearsal, because it is too expensive and time consuming to construct the models to support these applications. As a result, there is a critical need for faster and less costly techniques to build detailed 3-D models of complex real environments.

Current modeling techniques are interactive; a person either constructs 3-D sketches of an environment by looking at pictures of it or by applying photogrammetric techniques to inter-relate several images, measuring selected 3-D points, and then describing volumes and surfaces that capture the geometry of the scene. In both

cases, the process is tedious, which limits the level of detail incorporated into models, which in turn limits their utility.

The computer vision community has recently developed (and is still in the process of developing) techniques for automatically extracting 3-D information from a set of snapshots and from long sequences of images. This progress, however, is so new that the techniques have not been evaluated, except in a qualitative way.

Given this situation, we proposed to do the following:

1. Survey existing modeling techniques, from both the computer vision and photogrammetric communities
2. Construct an accurate ground-truth model of a complex building
3. Gather sets of controlled imagery of the site, which can be used for both developing algorithms and evaluating them
4. Explore ways to combine computer vision techniques for automatically extracting 3-D information from long sequences of images (such as video) with photogrammetric techniques for inter-relating large sets of images
5. Provide a systematic analysis and evaluation of the model construction techniques developed in (4)

Within our team, Calgis has extensive experience in surveying and photogrammetry and has developed a software package called UMENS that includes several techniques for interactively extracting information from one or more photographs. In addition, UMENS includes a generalized version of the photogrammetric bundle adjustment procedure that works with hundreds of images and supports constraints on scene features, such as requirements that a set of 3-D points lie on a vertical line or that a set of lines lie in a horizontal plane. In this project, Calgis is concentrating on tasks 1, 2, 3, and 5. SRI International has developed a set of techniques for identifying and tracking features through video sequences, plus techniques for sketching buildings from one or more images. In this project, SRI is concentrating on tasks 1, 4, and 5. The project has been under way for approximately 6 months. In this paper, we summarize our literature survey, and then briefly describe our approach, current status, and future plans.

2 Summary of Our Literature Survey

Video has many characteristics that make it a sensor of choice for some mapping applications. Among its advantages are low cost, real-time or near real-time operation, and data redundancy. In the past five years, video photogrammetry has begun to be used in architectural mapping [Streilein and Gaschen, 1995]. In general, it has been approached as traditional photogrammetry with accuracy ranging from 1/300 to 1/800 of the photographic distance.

The basic disadvantage of video imagery is its relatively low resolution compared to film. Standard video recorders have an image resolution of approximately 400 lines, compared to approximately 1500 lines for 35-mm slide film. In addition, video cameras generally have large lens distortions that complicate the mathematical models (and calibration) of the sensors, limiting their effective mapping accuracies.

Recently, the Global Positioning System (GPS) has been used for aerial mapping and mapping roads and railroad tracks from moving cars [El-Sheimy *et al.*, 1995]. For this project, we extended its use to ground-level architectural mapping from video sources.

At the beginning of the project, we made an extensive literature survey of techniques for extracting 3-D information from long monocular image sequences. This section is a short summary of that survey [Luong *et al.*, 1997].

The analysis of long monocular sequences has been known for many years in the computer vision community as the "structure-from-motion" problem. Only recently, however, have techniques been designed to take advantage of the redundancy afforded by a videotape as opposed

to a set of still photographs.

In this short survey, we describe the three primary approaches (minimization, recursive, and factorization) and mention a few techniques that do not fit into these classes. Note that we only consider methods designed specifically to handle multiple frames taken by a single camera – although stereo sensors have been applied to a few robotics applications, they are not widely available.

2.1 Minimization Methods

Minimization methods use optimization techniques to iteratively refine an initial estimate of the parameters. They typically are applied in batch mode and have the drawback that they may get stuck in a local minimum, depending on the starting value and the shape of the optimization surface. The computation is usually quite slow (a few minutes). On the positive side, many optimization techniques have been developed for specific tasks that converge quickly (for example, in less than a handful of iterations) and (almost) reliably to the optimal result.

Some methods first compute the motion, and then calculate the 3-D structure of the scene [Spetsakis and Aloimonos, 1991, Shariat and Price, 1990, Luong and Faugeras, 1997]. These techniques are based on the epipolar constraint, or a derived form of it, which makes it possible to eliminate the structure from the computation. The advantage is a reduction in the number of parameters to be minimized. The main drawback is that the epipolar constraint provides only one equation per point, since the constraint is in only one direction. This is unlike the equations based on the projection, which will be discussed next. However, it has been argued that not incorporating structure into the calculations results in less stable results only if global consistency across views is not enforced. The computation of a global representation requires taking into account scale factors between pairs of views.

A second class of method aims to recover structure and motion simultaneously. The idea is to perform a large-scale optimization, usually using the Levenberg-Marquardt (LM) minimization algorithm. The objective function to be minimized is the difference between the measured coordinates and the coordinates derived from the unknown points and projections. This is similar to the idea of "bundle adjustment" in photogrammetry. This method was combined with a motion model in [Broida and Chellappa, 1991]. It has regained interest within the computer vision community for dealing with uncalibrated images [Mohr *et al.*, 1993, Szeliski and Kang, 1994, Hartley, 1994]. These methods are good for enforcing all the constraints of the problem. They produce a 3-D reconstruction of the scene and an estimate of the motion. The principal drawback is that they can get trapped into local minima; therefore, a good initialization might be required. In addition, the computational requirement can be high, even if sparse methods are used to speed them up. The number of variables is at least (3*M + 6*(N-1)), if we assume that M points are observed in N views.

2.2 Recursive Methods

An algorithm is said to be recursive when the solution for frame K+1 is determined from the solution for frame K and that data at frame K+1, with no overhead as more frames are added. Old observations can also be discarded. These approaches are necessary if there is a requirement for real time. Even if it is not the case, these approaches can deal with a lot of data in a reasonable amount of time.

The problem is formulated by posing structure from motion as a parameter estimation problem. Almost all these approaches are based on the Extended Kalman Filter (EKF). Because of the nonlinear nature of the measurement and plant equations, the state estimates are not sufficient statistics for the past data (in contrast with the linear case and the Kalman Filter, which is optimal), so degradation can occur, for instance, if the initialization is not precise enough. For nonlinear optimization, the recursive methods are not optimal, in contrast to the batch methods [Weng *et al.*, 1993], although results close to optimal could be ob-

tained [McLauchlan and Murray, 1995]. On the other hand, if the models are only approximations to the true plant and measurement equations, then small deviations from the models are tolerated without explicit modeling, thanks to the "limited memory" of the system, which tracks the actual parameters and "forgets" about data taken at earlier times.

In [McLauchlan and Murray, 1995] it was claimed that better results are obtained with a complete parameterization of the structure (X,Y,Z), a partial representation Z (like in [Azarbayejani and Pentland, 1995]) being better than no structure (as advocated by Soatto et al). However, in [Soatto and Perona, Nov 1995] it is argued that it is not the inclusion of the structure that makes a scheme robust, but it is the formulation of the problem as a global model (which refers to a common reference, for instance that of the initial time instant, sometimes called "object-centered") that does it.

Few recursive methods decouple the estimation of motion from the estimation of structure. Apart from [Weng *et al.*, 1987], which is based on a particular motion model, this approach has been mostly advocated by Soatto and coworkers [Soatto *et al.*, 1994, Soatto and Perona, 1995, Soatto and Perona, in press 1996, Soatto and Perona, 1996]. If the feature points are available throughout the sequence, then it is advisable to use a global model (which refers to a common reference – for instance, that of the initial time instant). These models have been found to be more robust and precise, and are more easily implemented using the structure. The main problem with global models is that they cannot handle occlusions, in the sense that information for each point can be integrated in time (or across different frames) only to the extent that it is visible. In the case of a minimization approach, this can be taken care of by reweighting. In the case of a recursive approach, there is a transient from the initialization that affects the estimates of all other parameters. Therefore, there is a tradeoff: if feature points have a lifespan that is long enough (longer than the convergence rate of the optimization/filtering scheme, typically 10 to 20 frames) then it is better to use a global scheme. Otherwise, it might be better not to use a global scheme.

For the methods that compute structure and motion together, there are two broad approach classes. In the first, for each new image, the old shape estimate is combined with the information contained in the new image, which is considered as the measurement. Each new image only partially constrains the shape. Typical examples, where the extended Kalman Filter is used, are [Broida and Chellappa, 1990] (constant motion model) and [Azarbayejani and Pentland, 1995] (no explicit motion model). In [McLauchlan and Murray, 1995], which introduced the Variable State Dimension Filter, no motion model is used at all.

The alternative is to apply a two-frame algorithm for each new frame [Kumar *et al.*, 1989, Cui *et al.*, 1990, Oliensis and Thomas, 1991, Soatto *et al.*, 1993]. The measurement is the result of this algorithm. An advantage of this approach is that no model of the motion is required. A potential drawback is that if the interframe motion is too small, the result obtained with a two-frame algorithm can become unstable.

2.3 Factorization Methods

This class of methods assumes a linear projection (orthographic [Debrunner and Ahuja, 1990, Tomasi and Kanade, 1992], weak perspective [Weishall and Tomasi, 1995], paraperspective [Poelman and Kanade, 1994], affine with self-calibration [Quan, 1994]). It computes simultaneously general structure and motion using a simple and elegant scheme. A "measurement matrix" is factored in a product of structure and motion by using the fact that it has theoretical maximal rank 3. The advantages are

- No hypothesis is made on the motion, or on the structure.
- All the data in all the images are treated uniformly.
- Solution is via linear methods, either batch or incremental [Kanade and Morita, 1994, Weishall and Tomasi, 1995] (because of the

linearity of the equations, these two formulations are equivalent, unlike the nonlinear case). No initialization is required and convergence is guaranteed.

- Because of the linear camera model, there are very few intrinsic parameters. Usually, the knowledge of the aspect ratio is sufficient.

The drawbacks are

- The main problem is that the linearity of the projection is only an approximation. No real camera is affine, they are all perspective. The approximation is valid only within a certain domain, when the relative depths in the scene have a small variation with respect to the distance to the camera.
- The method works best only if every primitive is visible in every image.
- It is not possible to incorporate a full statistical error model (weighting according to uncertainty, outlier detection) beyond the implicit least-squares tradeoff.

2.4 Other Approaches

Several methods have been tried with the goal of reconstructing structure in a more efficient and reliable way than the bundle adjustment methods, by exploiting some particular properties of the projection, in order to partly linearize the problem.

In [Kumar *et al.*, 1992], the reconstruction of structure is performed in two steps to overcome the sensitivity to errors in relative orientation. In a first step, the scene is partly reconstructed based on "shallow structures," whose extent in depth is small compared to the distance to the camera.

The goal of [Oliensis, 1994] is to address the case where (a) the depth variation in the scene is large, and therefore linear methods are not applicable and (b) the baseline is small, and therefore methods that rely on only a few images are unstable. The algorithm first cancels the rotation by using linear techniques and then proceeds iteratively to compensate for the errors introduced by the approximations.

In [Christy and Horaud, 1994] an iterative method is provided to solve the bundle adjustment equations. The method requires only three to five affine iterations, which are linear. It is easier to analyze since the relation between perspective and weak perspective has been clarified.

An extension of the factorization methods is introduced for projective reconstruction in [Sturm and Triggs, 1996].

In [Faugeras *et al.*, 1995], a method is presented to perform a Euclidean reconstruction from multiple uncalibrated views, which can be taken by different cameras. In [Faugeras and Laveau, 1994], a technique is presented to generate novel views of a scene without the need to compute an explicit 3-D model. This method might be a good complement to the previous one, in cases where it is difficult to get geometric information from the scene.

2.5 The Berkeley Facade Project

The Facade Project [Debevec *et al.*, 1996] at the University of California at Berkeley is the closest system to our proposed sketch-first modeling technique. Facade is a system designed to model and render architecture from photographs. The modeling approach, which combines both geometry-based and image-based techniques, has two components. The first component is a domain-specific photogrammetric modeling method that facilitates the recovery of the basic geometry of the photographed scene, exploiting the constraints that are characteristic of architectural scenes. The user must completely specify a polyhedral model, as well as the model-to-image correspondences. The camera pose and the metric parameters of the model are recovered automatically. The second component is a model-based stereo algorithm, which recovers the local geometric deviations from the basic model.

Facade, however, was designed to work with

a set of snapshots, not video. As a result, it requires the user to hand mark all the correspondences, whereas in a video-based system, the temporal continuity could be exploited for automatic tracking of features. An additional benefit of using video would be to exploit the massive redundancy of information to gain robustness and precision over the use of a sparse set of photographs.

3 Ground-Truth Model

Calgis and SRI evaluated several candidate building sites before selecting an interesting building complex (136 x 47 x 19 m) for this project. Considerations that favored this selection included the availability of numerous distinctive structural features, such as building corners and window corners, for use as control points (thereby avoiding the need for placing special targets on the building); a diverse mix of rectangular and nonrectangular building components; a wide variety of building heights; and a set of unobstructed views of major portions of the building.

With the objective of creating a high-precision photogrammetrically mapped model for the building complex, two types of photographic coverage were acquired – aerial images and ground-level images. A wide-angle (152 mm focal length, 230 x 230 mm format) metric aerial camera was used for planimetric mapping of the building complex and its immediate surroundings. The building was covered by a pair of overlapping vertical aerial photographs and a pair of obliques.

To accurately map all faces of the building, a Wild P32 metric terrestrial camera (focal length 64.09 mm, 60 x 80 mm format) was used to acquire overlapping terrestrial photographs from a series of locations around the perimeter of the complex. Fifty photos were taken to provide full coverage.

Photogrammetric mapping requires some points with known 3-D object space coordinates, which can be identified in each set of overlapping photographs. Such control points are usually established through physical measurements carried out in object space. When a large set of overlapping photos that constitute a block are to be processed, the object-space measured control points are used to densify such control through block adjustment. Following this approach, field survey measurements were used to establish suitable points to control the aerial photographic model, and to provide sufficient control points, appropriately distributed over the external faces of the building, to permit a satisfactory bundle adjustment solution for the terrestrial photo block. In total, 150 photo ID points were established on the building with an accuracy of 0.03 m, using traditional surveying measurements. These points were tied to a network of 12 GPS control points that has an accuracy of 0.02 m. Since GPS surveying techniques usually provide somewhat poorer accuracy in the height component, differential leveling was used to establish the height component.

All photos were scanned at a resolution of 0.015 mm; 6189 line features were digitized to an accuracy of 0.05 m, using an Intergraph Imagestation Workstation and the UMENS software.

4 Video Imagery

The methodology for establishing the kinematic positioning of a GPS antenna in motion is well known. The fundamental need, in this project, however is to determine the spatial position of the video camera when each video frame is acquired. This poses two problems. First, the spatial offset between the video camera (perspective center) and the GPS antenna (phase center) must be determined and maintained (without changes) during the image collection phase. And second, the temporal relationship between the video frames and the antenna location must be established.

Trimble GPS receivers provide an option to output 1 timing pulse per second, synchronized with GPS time. They also output information to a message file that identifies the Universal Time (UT) corresponding to each pulse. Horita, Inc. markets an interface to Trimble receivers that generates timing signals to be added to the audio track of a videotape. These sig-

nals are synchronized with the GPS receiver's output timing pulses. They provide tags that can be used to identify the GPS location of the frame. This device also adds frame numbers (0 through 29) to tagging information. When such a time-tagged video tape is replayed, the time and frame number can be displayed with each frame. This commercial device was specially modified to interface the Sony camcorders to the Trimble 4000SSI GPS receiver. All 30 frames (0 to 29) shot within a second are marked with the same UT. Interpolation can be used to estimate the time of each frame.

This approach was successfully implemented for collecting data with a single video camera, as well as for a pair of genlocked stereo cameras, with a base line of 2.0 m. To date, we have taken three types of video sequences – stereo video with GPS data, monocular video with GPS data, and handheld video without GPS.

5 Sketch-First Modeling

A standard photogrammetry-based approach to constructing 3-D models from a set of images is to

1. interactively identify points that occur in two or more images
2. automatically compute the relative positions and orientations of the cameras that took the images, using photogrammetric techniques
3. automatically compute the 3-D locations of the selected points by intersecting the appropriate rays from multiple cameras
4. interactively identify model primitives, such as planes and lines, that occur in the scene
5. interactively construct 3-D models of objects in the scene by combining and extending the model primitives

This process is effective, but time consuming, because a person must identify numerous of matching points, and then construct the model from the computed 3-D points. If video data are provided, instead of snapshots, an additional step would be inserted before step 1. The user would interactively select a set of "key frames" from the video, and then use them as the set of images.

Computer vision techniques have the potential for reducing the involvement of a person in the three interactive steps in this process. First, computer vision techniques could perform step 1 automatically, or almost automatically. They could select and match the points needed to compute the relative locations of the cameras. Video data would simplify this process because the temporally coherent nature of the image sequence facilitates the tracking of points from image to image, which could establish matches from one key frame to another. In fact, the system could automatically select key frames based on the amount of overlap between new frames and old ones. Second, computer vision techniques could perform the segmentation and modeling steps, numbered 4 and 5. Given a cloud of points and/or line segments, the techniques would segment the features into planes and extended lines, and then fit planes and lines to the associated features, being careful to throw out features that do not belong.

In our experience, current computer vision techniques are better at selecting, matching, and tracking features than in segmenting clouds of features and fitting 3-D structures to them. Therefore, in the near term, we foresee a person doing the bulk of the segmentation and modeling. If that is the case, then we propose that the whole process could be simplified by having the person first draw a quick sketch of the model components and then identify a few matching features in the images. The program could use the sketch to guide the rest of the processing.

We propose to develop a sketch-first modeling technique that is based on this line of reasoning. It begins by having the user define the basic elements of the model, state constraints between them, and identify a few matching features in the images. We envision a sketch vocabulary that includes "points," "lines," "planes," and "rectangular solids" as feature primitives, "hor-

izontal" and "vertical" as unary properties, and "in a plane," "perpendicular," and "coplanar" as binary relationships between primitives.

Note that the sketch is not necessarily a complete explicit model of the building, nor does it have to be metrically accurate. For our purposes, since we are using the UMENS bundle adjustment software, it can be a list of constraints on features. Thus, our sketch-first approach could work with a variety of sketch types, depending on the underlying solution procedures. At one extreme, the system could work directly with parameterized constructive solid geometry models, and at the other extreme, it could work with lists of constraints derived from separate images.

There are several potential benefits of starting with a sketch. First, the sketch defines key building primitives and associates names with them, either implicitly or explicitly. Second, it provides a way to specify 3-D constraints to observed features. For example, a user can specify that one line is horizontal and another is perpendicular to it. Third, it can help keep track of the portions of the building that have been imaged. Fourth, it can help identify matching mistakes by locating points that are not consistent with plane-to-plane transformations. Fifth, it can help identify points that are not on the building, such as points on vegetation.

5.1 The Technique

The high-level steps in our proposed sketch-first modeling technique are

Instantiate the Data Set – select the images and optionally estimate the parameters of the associated cameras.

Generate a Sketch – select building primitives, identify a few occurrences of matching features in the images, and specify constraints on the primitives constraints.

Compute the 3-D Geometry – apply photogrammetric techniques to estimate the locations of the cameras and the building primitives, simultaneously.

Complete the Model – edit the model, select image patches for rendering the surfaces, and generate crude models of vegetation.

Our approach to implementing these steps is as follows:

Instantiate the Data Set –

1. Interactively select relatively few images to be used for sketching the building. A person examines the available data (snapshots, aerial images, videos, and maps) and selects ones that cover the building, preferably in an overlapping fashion so that key features are visible in two or more images. For video sequences, an automatic technique can identify features, track them, and select key frames that guarantee a pre-specified amount of overlap between frames.
2. (Optional) Interactively estimate the "camera parameters" for the selected images and maps. If the parameters are known from meta data associated with the imagery, use them. If not, use a combination of contextual information and single-image and multiple-image techniques to estimate the parameters. For example, if a trihedral building corner is visible in an uncalibrated image, use sets of parallel lines on the planes meeting at the corner to estimate the internal parameters, such as focal length and piercing point of the camera.

Generate a Sketch – Interactively construct a sketch of the building. This is usually performed by analyzing one aspect of the building at a time. The person generates lists of features and constraints for one aspect, and then moves on to another one, tying them together by identifying common features or by adjusting primitives to agree with multiple views.

1. Automatically generate lists of information-rich (i.e. interesting)

features, such as points and lines, in the imagery of the current aspect. (We have applied "interest operators" to locate distinctive points and edge-detection and line-fitting algorithms to locate prominent line segments.) If key features are missed by the automatic techniques, they can be added interactively.

2. Interactively define a local coordinate system for this aspect. One way to do this is to select detected line segments that define a rectangular trihedral vertex. For some buildings one coordinate system is sufficient for the whole building. For others, multiple coordinate systems may be required, if they have sections that meet at odd angles.

3. (Optional) Interactively outline regions in the images that are covered by vegetation, such as trees and bushes. These outlines can save time by limiting the application of automatic analysis techniques to regions that are appropriate. In addition, they can be used at the end of the modeling process to construct crude volume models of the vegetation.

4. Interactively sketch a set of building components by selecting primitives, such as a rectangular solid, and then roughly aligning them with detected features in one image. For example, after selecting a rectangular block to represent the basic shape of the building (aligned with the selected local coordinate system, by default), move it and resize it so that it is approximately positioned correctly in the image. With approximate camera models for the images, primitives can be interactively sized in 3-D to match several images. Without approximate camera models, but with the images taken from a video sequence, use the tracking results through the sequence to automatically identify corresponding features. If there are no camera models and the imagery is an unrelated set of pictures (i.e., not video), interactively point at a few features of the solid, such as its edges and faces, and identify them in the other images.

5. Interactively add details, such as a row of windows, to the sketch by selecting the plane of the windows, drawing one window, and then specifying the pattern on the plane. The system can automatically recognize occurrences of a sketched feature, such as a window, by adjusting its size and shape as it is moved over the plane, and then identifying matching lines. If vegetation regions have been specified (in step 3d), skip those areas. This technique can quickly populate many details on a basic model.

Compute the 3-D Geometry –

1. Semiautomatically select points along the line segments detected in the images and identify their corresponding points in other images. UMENS currently only works with points and constraints on them, not lines. Therefore, we need to select specific points along the lines and add a constraint that they lie on a 3-D line. If there are naturally occurring distinctive points along the line caused by such things as window frames or wall textures, they can be used as specific 3-D points. If not, there are techniques for constructing corresponding points in two views of a planar surface. For example, if we have identified two parallel line segments that define a rectangle in the world, the image locations of their end points (if visible) can be used to generate a set of corresponding points along the lines in the images.

2. Automatically compute the locations of the cameras and the locations of the feature points by applying the UMENS bundle adjustment proce-

dure, which computes the resection of all the cameras.

3. Automatically compute the 3-D locations of the feature points by intersecting the rays from the resected cameras.
4. Automatically fit lines and planes to the 3-D feature points associated with the line and plane features. Eliminate points that are gross errors.

Complete the Model –

1. Interactively edit the computed model by examining its predicted appearance in several images.
2. Select image patches to be used for rendering the 3-D models. If the vegetation regions have been specified, this can be done automatically, selecting image patches that are as free of vegetation as possible. If not, interactively select the image (or images) for extracting the appropriate patches.
3. (Optional) Construct volumetric models of the vegetation.

5.2 Current Status

To date we have made the following progress on the steps described above:

Instantiate the Data Set – We have developed an automatic technique for selecting interesting point features in individual video frames; track them from frame to frame; estimate the overlap of the current image with previous images; select key frames, when the overlap drops below a specified amount; and incrementally add interesting points in new image regions. To do this we have developed a blackboard mechanism that keeps track of the portions of the scene that have been viewed by the video sequence. Our blackboard is similar to a mosaicked image, except that our primary purpose is to identify "tie" points for inclusion in the bundle adjustment procedure. Therefore, whenever a new image overlaps a previous image, whether it was the immediately preceding image or an image much earlier in the sequence, we can identify previously tracked points and predict their locations in the current images. If we can find them in the new image, we add the matching locations and image numbers to their list of matches.

In addition, we have developed an automatic technique for checking and refining tracked feature points by comparing the frame-to-frame results with independent matches produced by "hopping" over several frames. We found that the image-to-image matches drift slowly and occasionally make gross mistakes. Thus, if we initially see a feature in image 17 and track it through image 62, it may have drifted 2 or 3 pixels from the point that a person would have picked. To minimize this drift, we periodically make matches over several images, for example, image 17 to image 37. If the image-by-image match is close to the "hop" match, we reset the feature's location to the hop match's location. If the matches are quite different, we mark the point as untrustworthy.

We have experimented with UMENS techniques for interactively estimating camera parameters for images that do not have meta data. We have used sets of parallel lines to estimate focal length and the sizes of known objects to establish scale.

Generate a Sketch – We have interactively constructed sketches from one or more images. We used the RCDE to define the local coordinate systems, and then to populate a scene with building components that were aligned with that system.

Compute the 3-D Geometry – We generated a list of a few techniques for constructing points along sets of lines viewed in two perspective images.

We interactively computed the camera locations associated with several images, using the UMENS bundle adjustment software, and then automatically computed

the 3-D locations of feature points (using UMENS software).

In addition, we have experimented with techniques for robustly fitting planes to sets of 3-D points.

Complete the Model – We interactively edited a set of 3-D models, using the RCDE System.

6 Characterization and Evaluation of the Methods

We plan to evaluate the new technique along two dimensions, the amount of effort required to construct a detailed 3-D model and the accuracy of the final result. In addition, we plan to evaluate the utility of incorporating individual components, such as automatic feature tracking, into the UMENS modeling system.

These evaluations, particularly the amount-of-effort-based measurements, are difficult because there are user training issues, computer speed issues, and scene complexity issues. The evaluation of the final model is more straightforward, but geometric accuracy is not the only aspect that determines the utility of a model for such tasks as simulation and mission rehearsal. If it takes months to construct, the crisis may have passed before a model can be built.

Calgis is compiling a detailed list of man-hours required to construct the ground-truth model from aerial images, ground survey points, and metric photographs. This level of effort will be used as an estimate of the time required by current techniques. In June, when we use the sketch-first modeling technique to model the same building, we will document the amount of time required for each step and compare it to Calgis's list of man-hours.

The Calgis 3-D geometric model will be substantially more accurate than we expect to be able to generate from video data. (Calgis expects that the absolute geo-referenced locations of building features, such as the corners of windows, will be mapped to within 5 cm.) As a result, we plan to use the Calgis model as ground truth. We will characterize the geometric errors in a proposed building model by computing statistics on the differences between the ground truth locations and the model's locations.

7 Future Plans

We plan to make portions of the ground truth model and experimental data available to anyone interested in working with it.

With respect to the sketch-first modeling technique, our plans are to complete an initial implementation by June and use it to model the building for which Calgis has constructed a ground-truth model. We will characterize the effort required and the accuracy of the result. This initial implementation will be a combination of interactive and automatic steps that represents a step toward a faster and less costly system for constructing detailed 3-D models of real scenes.

References

[Azarbayejani and Pentland, 1995] A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):562–575, 1995.

[Broida and Chellappa, 1990] T.J. Broida and R. Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *IEEE T-AES*, 36(4):639–656, 1990.

[Broida and Chellappa, 1991] T.J. Broida and R. Chellappa. Estimating the kinematics and structure of a rigid object from a sequence of monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):497–513, 1991.

[Christy and Horaud, 1994] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. Technical Report RR-2421, INRIA, December 1994.

[Cui *et al.*, 1990] N. Cui, J.J. Weng, and P. Cohen. Extended structure and motion analysis

from monocular image sequences. In *Proc. International Conference on Computer Vision*, pages 222–229, 1990.

[Debevec *et al.*, 1996] P. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs. Technical Report UCB/CSD-96/893, University of California, Jan 1996. derived papers in ECCV and Siggraph.

[Debrunner and Ahuja, 1990] C. Debrunner and N. Ahuja. A direct data approximation based motion estimation algorithm. In *Proc. International Conference on Pattern Recognition*, pages 384–389, 1990.

[El-Sheimy *et al.*, 1995] N.M El-Sheimy, K.P. Schwartz, and M. Gravel. Mobile 3-d positioning using gps/ins/video cameras. In *1995 Mobile Mapping Symposium, ASPRS*, Columbus, Ohio, May 1995.

[Faugeras and Laveau, 1994] Olivier Faugeras and Stéphane Laveau. Representing three-dimensional data as a collection of images and fundamental matrices for image synthesis. In *Proc. International Conference on Pattern Recognition*, pages 689–691, Jerusalem, Israel, 1994.

[Faugeras *et al.*, 1995] Olivier Faugeras, Stéphane Laveau, Luc Robert, Cyril Zeller, and Gabriella Csurka. 3-d reconstruction of urban scenes from sequences of images. In A. Gruen, O. Kuebler, and P. Agouris, editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pages 145–168, Ascona, Switzerland, April 1995. ETH, Birkhauser Verlag. also INRIA Technical Report 2572.

[Hartley, 1994] R.I. Hartley. An algorithm for self calibration from several views. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 908–912, Seattle, WA, 1994.

[Kanade and Morita, 1994] T. Kanade and T Morita. A sequential factorization method for recovering shape and motion from image streams. In *Proc. DARPA Image Understanding Workshop*, pages 1177–1187, Monterey, California, 1994.

[Kumar *et al.*, 1989] R. Kumar, A. Tirimalai, and R. Jain. A non linear optimization algorithm for the estimation of structure and motion parameters. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 136–143, 1989.

[Kumar *et al.*, 1992] R. Kumar, H. Shawney, and A.R. Hanson. 3d model acquisition from monocular image sequences. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 209–215, 1992.

[Luong and Faugeras, 1997] Q.-T. Luong and O.D. Faugeras. Self calibration of a moving camera from point correspondences and fundamental matrices. *Intl. Journal of Computer Vision*, 22(3), 1997.

[Luong *et al.*, 1997] T. Luong, R. Munjy, and R.C. Bolles. Structure from long monocular image sequences: A survey. Technical report, SRI International and Calgis, Inc., 1997.

[McLauchlan and Murray, 1995] P.F. McLauchlan and D.W. Murray. A unifying framework for structure and motion recovery from images sequences. In *Proc. International Conference on Computer Vision*, pages 314–320, Cambridge, Ma, 1995.

[Mohr *et al.*, 1993] R. Mohr, F. Veillon, and L. Quan. Relative 3d reconstruction using multiple uncalibrated images. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 543–548, NYC, 1993.

[Oliensis and Thomas, 1991] J. Oliensis and J. Thomas. Incorporating motion errors in multiframe structure from motion. In *IEEE workshop on visual motion*, pages 8–11, Princeton, NJ, 1991.

[Oliensis, 1994] J. Oliensis. A linear solution for multiframe structure from motion. In *Proc. DARPA Image Understanding Workshop*, pages 1225–1231, Monterey, California, 1994.

[Poelman and Kanade, 1994] C.J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. In *Proc. European Conference on Computer Vision*, pages B 97–108, Stockholm, Sweden, 1994.

[Quan, 1994] L. Quan. Self-calibration of an affine camera from multiple views. Technical Report RT 125 IMAG - 26, LIFIA, Dec 1994. To appear in IJCV.

[Shariat and Price, 1990] H. Shariat and K. Price. Motion estimation with more than two frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):417–434, 1990.

[Soatto and Perona, 1995] S. Soatto and P Perona. Dynamic rigid motion estimation from weak perspective. In *Proc. International Conference on Computer Vision*, pages 321–328, Cambridge, Ma, 1995.

[Soatto and Perona, 1996] S. Soatto and P. Perona. Motion from fixation. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 817–824, San Francisco, CA, 1996.

[Soatto and Perona, in press 1996] S. Soatto and P. Perona. Recursive 3-d visual motion estimation using subspace constraints. *Int. J. of Computer Vision*, in press, 1996.

[Soatto and Perona, Nov 1995] S. Soatto and P. Perona. Reducing "structure from motion" 2: experimental evaluation. *submitted to the IEEE trans. PAMI*, Nov. 1995. short version in the proc. of the CVPR 96.

[Soatto *et al.*, 1993] S. Soatto, P. Perona, R. Frezza, and G. Picci. Recursive motion and structure estimation with complete error characterization. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 428–433, New-York, NY, 1993.

[Soatto *et al.*, 1994] S. Soatto, R. Frezza, and P. Perona. Motion estimation on the essential manifold. In *Proc. European Conference on Computer Vision*, pages B–61–72, Stockholm, Sweden, 1994.

[Spetsakis and Aloimonos, 1991] M. Spetsakis and Y. Aloimonos. A multi-frame approach to visual motion perception. *The International Journal of Computer Vision*, 3(6):245–255, 1991.

[Streilein and Gaschen, 1995] A. Streilein and Stephen Gaschen. Close range techniques & machine vision. In *ISPRS, Vol. XXX, Part 5*, Melbourne, Australia, March 1995.

[Sturm and Triggs, 1996] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proc. European Conference on Computer Vision*, pages II–709–720, Cambridge, UK, 1996.

[Szeliski and Kang, 1994] R. Szeliski and S.B. Kang. Recovering 3d shape and motion from image streams using nonlinear least squares. *JVCIR*, pages 10–28, 1994.

[Tomasi and Kanade, 1992] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *The International Journal of Computer Vision*, pages 137–154, 1992.

[Weishall and Tomasi, 1995] D. Weishall and C. Tomasi. Linear and incremental acquisition of invariant shape models from images sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):512–517, 1995.

[Weng *et al.*, 1987] J.J. Weng, T.S. Huang, and N. Ahuja. 3-d motion estimation, understanding and prediction from noisy image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(3):370–389, 1987.

[Weng *et al.*, 1993] J. Weng, N. Ahuja, and T.S. Huang. Optimal motion and structure estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):864–884, 1993.

Visibility Estimation from a Moving Vehicle Using the RALPH Vision System

Dean Pomerleau
Robotics Institute, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh PA 15213
pomerlea@cs.cmu.edu
http://www.cs.cmu.edu/~pomerlea

Abstract[1]

Reduced visibility is a common casual factor in many traffic accidents. This paper describes a forward looking vision system which simultaneously track the lane and estimate visibility. The system estimates visibility by measuring the attenuation of contrast between consistent road features at various distances ahead of the vehicle. Results of experiments on simulated images, as well as live vehicle tests are presented.

1. Introduction

Reduced visibility caused by fog, rain, snow, darkness and glare is a frequent contributing factor to traffic accidents [Allport, 1989]. In fact, some of the most serious of all highway incidents, sometimes involving dozens or even hundreds of vehicles, occur when reduced visibility conditions result in a chain reaction of crashes. Paradoxically, some advanced technology, like Adaptive Cruise Control (ACC) systems have the potential to decrease, rather than increase safety in these situation by encouraging drivers to travel at a speed and headway distance that may not be safe for the ambient environmental conditions. This paper describes the first step in the solution to this problem, a system that can estimate the ambient visibility from a moving vehicle.

1. This research was sponsored by Office of Naval research (ONR) under Contract N00014-95-1-0591. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ONR or the U.S. Government.

There are several technologies typically employed to estimate visibility, including transmissometers, which measure the transmittance of the atmosphere over a baseline distance, and nephelometers which measure the scattering coefficient of an air sample caused by suspended particles [National Weather Service, 1996]. Unfortunately, these systems suffer from several drawback for automotive applications. Transmissometers require a transmitter and a receiver a substantial distance (typically hundreds of meters) apart, which is very difficult to implement on a moving vehicle. Stationary transmissometers located near stretches of roadway commonly plagued with poor visibility can be effective for a local area, but may miss nearby reduce visibility conditions because of the very localized nature of some reduced visibility phenomena.

Nephelometers can be mobile, since they use a collocated transmitter and received to measure the backscatter of light off particles in the air. However they are prone to miss many of the important phenomena effecting how far a driver can truly see. These phenomena include:

- Opacity of the atmosphere due to particulates
- Ambient lighting conditions - sun, moon, overhead lights, direction of lighting
- Headlights from the driver's own vehicle and other vehicles
- Windshield transmissive properties due to dirt, water, snow or ice buildup.

The only way to automatically estimate the cumulative influence of these factors on the driver's abil-

ity to see potential obstacles ahead is to employ a sensing system which reasonably match the driver's perceptual characteristics. The system described in this paper accomplishes this match by using a CCD video camera pointing out the windshield of the vehicle, and processing the same features as the human driver to estimate visibility.

2. Approach

Manual visibility estimates are typically made by attempting to detect high contrast targets at various known distances. The farthest distance at which a target can be reliably detected is considered the visibility distance. Ideally, an automated visibility estimation system should work the same way. Unfortunately, it is very difficult to consistently find high contrast targets at various known ranges from a moving vehicle. Even the features that are supposed to be consistent on a roadway, the lane markings, vary greatly in their appearance, and are in fact frequently missing or obscured. The Rapidly Adapting Lateral Position Handler (RALPH) system [Pomerleau et al., 1996] overcomes this difficulty when detecting the position and curvature of the road ahead in camera images by utilizing whatever features are visible on the roadway, including lane markings, road/shoulder boundaries, tracks left by other vehicles, and even subtle pavement discolorations like the oil stripe down the lane center when necessary.

The visibility estimation system described in this paper exploits RALPH's ability to find and track arbitrary road features. In short, the system estimates visibility by measuring the attenuation of contrast between consistent road features at various distances ahead of the vehicle.

2.1. Road Feature Detection

To measure contrast between consistent road features, first these features must be detected in images of the road ahead. The algorithm the RALPH system uses to find road features is based on the observation that when viewed from above, a road resembles a ribbon of parallel bands formed by lane markings and other road features. To exploit this characteristics, RALPH first extracts from the image a trapezoidal region of the road ahead (See Figure 1). RALPH automatically varies

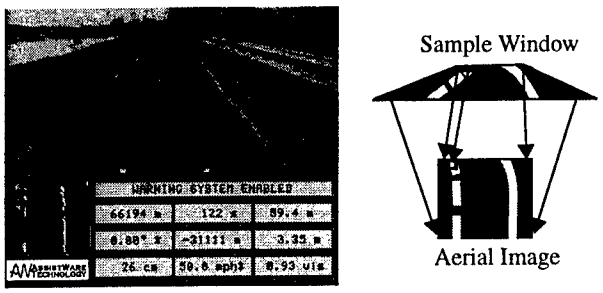


Figure 1: Forward looking image (left), and RALPH's sampling strategy (right).

the position of this trapezoid based on the vehicle's velocity and the current visibility, but under good conditions the top of the trapezoid is typically viewing the road between 50m and 120m ahead of the vehicle. RALPH resamples the image from this trapezoid. The horizontal extend of the trapezoid is set so that its width on the ground plane is identical at each row of the image. The horizontal distance that each row of the trapezoid encompasses is approximately 7.0 meters, about twice the width of a typical lane. This trapezoid is selectively sampled according to the strategy depicted in the schematic on the right of Figure 1 to create an aerial view of the road ahead. This sampling process results a low resolution (35x50 pixel) image in which important features such as lane markings, now appear parallel in the low resolution image (see schematic aerial view in the lower right of Figure 1, and the actual aerial view show in the lower left of Figure 1). Note that this image resampling is a simple geometric transformation (based on the assumption that the road is locally planar), and requires no explicit feature detection.

RALPH then uses this aerial image to locate the road ahead. To accomplish this, RALPH uses a one-dimensional representation of the road, created by taking a cross section of the aerial image perpendicular to the road, called the road template. The aerial image for the road in Figure 1 and road template created from a cross sections at the bottom of the image, are shown in Figure 2

There are several things to note about the template cross section. First, the lane markings show up quite distinctly as the two highest peaks. Also apparent in the cross section are two sharp dips just outside the lane markings, caused by a black filled seam in the pavement on the left side of the lane,

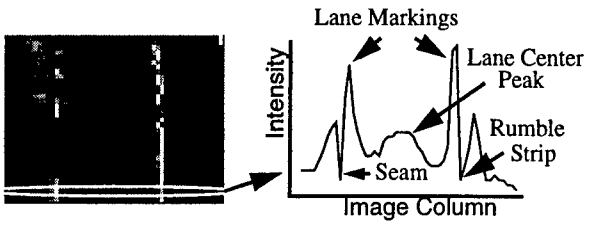


Figure 2: An aerial road image (left) and cross sections taken from the bottom of the image (right).

and the dark banding of a rumble strip on the right side. Finally, down the center of the lane the pavement is slightly lighter in intensity than the more heavily worn pavement closer to the lane boundaries, causing a wide shallow peak in the center of the cross section.

RALPH exploits all of these features to find the road ahead by using the entire one-dimensional cross section as a template. For each row of the aerial image, RALPH shifts the template left or right until it best matches the particular row's cross section. The amount of shift required to match a particular row is proportional to the lateral displacement of the lane center at that row of the image. For more details on the algorithm RALPH employs to generate and maintain the template, and how RALPH finds the position and curvature of the road ahead using the template, see [Pomerleau et al., 1996].

2.2. Visibility Estimation

In order to estimate visibility, the system uses the shifted road cross sections generated during the road detection process. Two such cross sections, one from the top of the aerial image, and one from the bottom, are shown in Figure 3. Notice that at the top of the image, relative far ahead of the vehicle, the peaks in the cross section are not quite as high, and the dips are not quite as low as the at the bottom of the image, close ahead of the vehicle. Qualitatively, it is this attenuation of contrast between features with increasing distance from the vehicle that the visibility estimation algorithm (described below) is measuring.

To quantify the feature attenuation, the system estimates for several rows at the top and bottom of the image, the median intensity around the lane center,

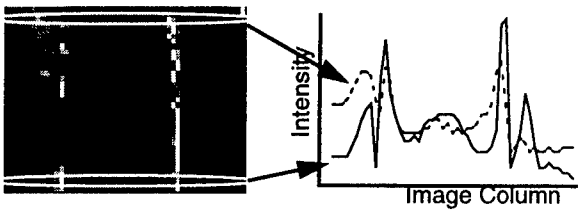


Figure 3: Road cross sections from top (dashed curve) and bottom (solid curve) of the aerial

as well as the maximum deviation from this median intensity within the row. The system averages the maximum intensity deviation for the rows at the top, and the rows at the bottom of the image, to overcome the effects of intermittent dashed lane boundaries and other image artifacts. The difference between the average maximum intensity deviation at the bottom and the top of the aerial image is the system's estimate of contrast attenuation.

In order to estimate visibility, it is not enough to simply measure contrast attenuation, since visibility should be a function of distance. Therefore, the contrast attenuation as measured above is scaled based on the distance between the top and bottom of the RALPH's view trapezoid (which can vary as mentioned previously). The resulting value is a measure of contrast attenuation per meter.

The final step in estimating visibility is normalization. Even under clear conditions like that shown in Figure 1, the contrast in the aerial image is significantly attenuated, even over the relative short distance between the bottom and the top of the image (see Figure 3). This is cause primarily by imaging artifact relating to the pixel spacing on the CCD array, and the camera's limited depth of field. Together these artifacts result in a blurring towards the top of the aerial image under all conditions. To eliminate the effect of this blurring on the visibility estimate, the contrast attenuation per meter value is normalized, so that the rate of attenuation on a bright clear day is equivalent to a visibility of 1.0, and visibility under degraded conditions are expressed relative to this baseline.

Figure 4 depicts an example of a reduced visibility condition, night driving. In this situation, the driver's visual range is reduced due to the limited range of the vehicle's headlights. This can be seen in the reduced contrast towards the top of the view

trapezoid. Cross sections from the top and bottom of the aerial image for this night image are shown on the right of Figure 4. Note how the absolute intensity of the cross section, as well as the maximum contrast in the cross section, are greatly reduced towards the top of the image when compared with results from the daytime scene shown in Figure 3. As a result of the greater feature attenuation, the visibility for this situation, as computed with the algorithm described above, has dropped to33% of the clear daytime visibility (reported as "0.33 vis" in the lower right corner of Figure 4).

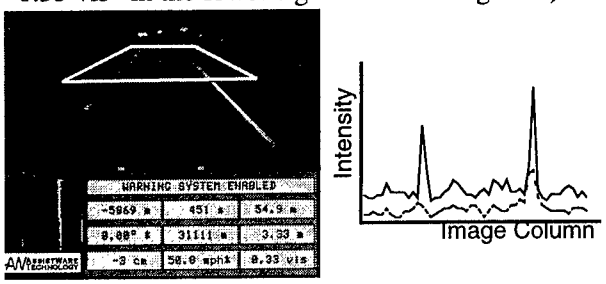


Figure 4: Night scene (left) with cross sections (right) from top (dashed) and bottom (solid) of the aerial image

3. Results

Two sets of experiments were conducted to test the visibility estimation algorithm's performance under a wide range of conditions. The first set of tests involved running the algorithm on a sequence of real road images in which various levels of simulated fog had been introduced through image manipulation. The second set of experiments involved live on-road tests of the visibility estimation algorithm.

3.1. Simulated Fog Experiments

As part of a project to test lane tracking systems under reduced visibility conditions [Pomerleau et al., 1995], Battelle Memorial Institute previously generated a set of images depicting various levels of fog from an image sequence collected on Carnegie Mellon's test vehicle, using Battelle's Electo-Optical Visualization and Simulation Tool (EOVAST) software. Given an original image, and accompanying estimates of camera characteristics, scene geometry and lighting conditions, the EOVAST software generates degraded versions of the same image as they would appear under user specified adverse weather conditions. The EOVAST software was originally developed for military targeting applications, and has been extensively validated for accuracy. For more details on EOVAST, and the results of the lane tracking tests under reduced visibility conditions see [Pomerleau et al., 1995].

In total, EOVAST was used to generate 120 reduced visibility images from 30 original images. These images depicted an interstate highway under foggy conditions with 700, 400, 300 and 100 meter visibility. A single one of the 30 original image, along with the same image in each of the four reduced visibility conditions is shown in Figure 5. These 150 images (30 original + 120 fog) were used to test the visibility estimation algorithm. Figure 6 shows the mean and standard deviation of the algorithm's visibility estimates for each of the five visibility conditions.

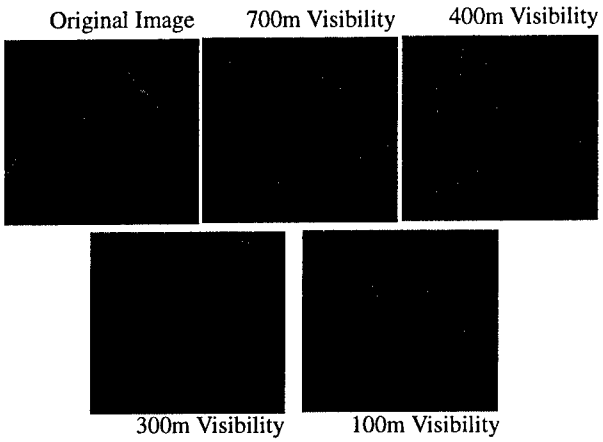


Figure 5: Original Image and four versions of the same image with simulated fog.

The first important characteristic of Figure 6 to notice is the substantial reduction in the algorithm's estimated visibility as the degree of fog increases (and hence the simulated visibility decreases). The second important attribute of Figure 6 is the large standard deviation in the algorithm's visibility estimates at each fog level (shown as the large spread in the error bars). Automatic visibility estimation with the algorithm reported here is a statistical process, since local variations in the underlying image features used to compute visibility can mask the contrast attenuation caused by ambient environmental factors. Therefore a relatively large number of images (more than 30) is required to determine visibility with certainty.

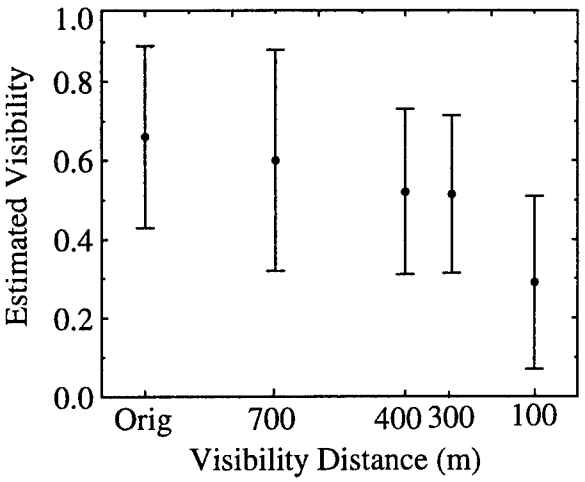


Figure 6: Mean and standard deviation of visibility estimates for the original image set, and the four reduced visibility conditions.

4. On-road Experiments

To overcome the problem of limited image data, and to test the algorithm under realistic conditions, a set of in-vehicle experiments were conducted using Carnegie Mellon's Navlab 8 test vehicle. Navlab 8 is an Oldsmobile Silhouette minivan equipped with a black and white video camera mounted behind the rear view mirror pointed through the windshield, and a Pentium-100 processor executing both the RALPH lane tracking algorithm and visibility estimation algorithm in real-time (15 frames per second).

Data on the visibility estimation algorithm's performance was collected on a 15 mile stretch of interstate highway, which offers several pavement types (concrete and asphalt) as well as a variety of lane delineating techniques, including solid and dashed white lane markings, yellow lane markings, retroreflectors, and roadside rumble strips. Data was collected on this stretch of roadway under six different conditions (See Figure 7 for example images from each condition):

- Daytime in good weather in the right lane
- Daytime good weather in the left lane
- Daytime in rainy weather
- Early morning with glare from the rising sun
- Nighttime with overhead lighting
- Nighttime without overhead lighting

The morning glare and the nighttime with overhead

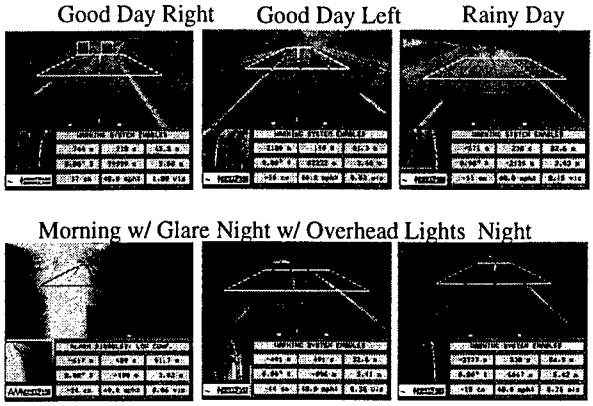


Figure 7: RALPH tracking the lane under various conditions, and estimating visibility.

lighting conditions occurred on only limited stretches of the 15 mile test road. Therefore the results reported below for these two conditions were compiled over only two and three miles of testing, respectively.

Figure 8 shows the results of the experiments on the six conditions, in decreasing order of estimated visibility. First note the visibility estimates in the left and right lanes in good daytime conditions were nearly identical to each other, and were far above the estimates for the other conditions. The next best visibility was reported for the nighttime with overhead lights condition. As can be seen from Figure 7, the overhead lights increase the range at which the road features are discernible, resulting in a corresponding increase in estimated visibility.

The nighttime condition with only headlight illumination was the situation the algorithm estimated to have the next best visibility, equivalent to approximately 30% of the good daytime visibility. Daytime rain, with significant water buildup on the windshield and substantial suspended spray in the air was determined by the algorithm to be the next to worse visibility condition tested. As Figure 7 shows, it is quite a bit more difficult to detect the road features, as well as other vehicles in this situation. However the lowest estimated visibility of the six tested was in the early morning glare condition. As is apparent in Figure 7, specular reflections off the pavement obscured the road features, and the very high ambient brightness saturated the camera, making it extremely difficult to detect the road (or anything else) anywhere except directly in front of the vehicle.

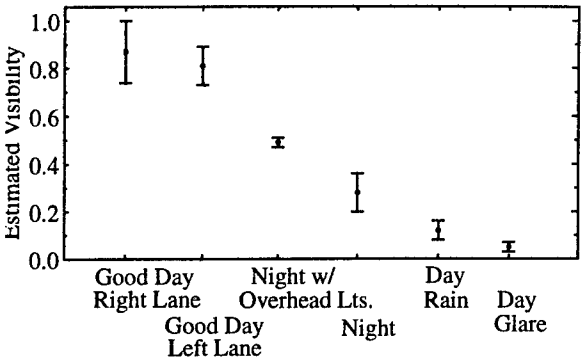


Figure 8: Mean and standard deviation of visibility estimates for the original image set, and the four reduced visibility conditions.

5. Discussion

The visibility estimation algorithm presented in this paper appears to perform well under a wide variety of conditions. The rank ordering of six conditions tested corresponds reasonably well to ones intuitive notion of how difficult it is to see in these situations. Note that traditional instruments for estimating visibility, which only detect suspended particles in the atmosphere, would have report less than unlimited visibility in only one of the six conditions tested, daytime rain. Interestingly, it is the very property for which vision systems are often criticized, their reduced effectiveness in adverse environmental conditions, which gives the algorithm its power. This is because the conditions in which the vision system has trouble seeing features are the same ones in which people have difficulty seeing.

One potential drawback of the visibility estimate technique presented is that it provides only a relative visibility measure, and not an absolute estimate of how far ahead road features or obstacles can be detected. However for a reduced visibility warning system, or a system to adjust the set speed and following distance of an adaptive cruise control, a consistent relative visibility measure may be sufficient. If an absolute measure of detection distance is required, it should be possible to calibrate the relative visibility estimates provided by the algorithm, although this hypothesis remains to be tested.

Live vehicle tests in fog still need to be conducted (fog is rare in Pennsylvania, particularly during the winter when these experiments were done). However, the results from the simulated fog experiments, and the live daytime tests in rainy conditions suggest that the algorithm should perform well, and report significantly reduced visibility under foggy conditions. Another possibility would be to combine this visibility estimation technique with a multispectral imaging device. By testing the visibility at different wavelengths, it may be possible to select the best wavelength(s) for operation under the current conditions.

6. Acknowledgments

The author gratefully acknowledges Larry Lazofson, Ed Kopala, and Howard Choe from Battelle for their hard work in generating the simulated fog images used during testing. This work was supported in part by the National Highway Traffic Safety Administration (NHTSA) under contract DTNH22-93-C-07023, by the Office of Naval Research (ONR) under contract N00014-95-1-0591, and by the Federal Highway Administration (FHWA) under cooperative agreement DTFH61-94-X-00001 as part of the National Automated Highway System Consortium. The content of this paper does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

References

[Allport, 1989] Najm, W., Mironer, M. and Fraser, L. (1995) "Analysis of Target Crashes and ITS Countermeasure Actions". *Proc. of 1995 ITS America Annual Meeting,* pp. 931-940.

[Pomerleau et al., 1995] Pomerleau, D., Kumar, B., Everson, J., Kopala, E., and Lazofson, L. (1995) "Run-Off-Road Collision Avoidance Using IVHS Countermeasures: Task 3 Report - Volume 1" NHTSA Contract DTNH22-93-C-07023.

[Pomerleau et al., 1996] Pomerleau, D. and Jochem, T. (1996) Rapidly Adapting Machine Vision for Automated Vehicle Steering. *IEEE Expert, Vol. 11, No. 2.* pp. 19-27.

[National Weather Service, 1996] "*Surface Weather Observations and Reports*" (1996) Federal Meteorological Handbook, 5th Edition, National Weather Service Publication FMH-1.

A Rapidly Adapting Machine Vision System for Automated Vehicle Steering

Dean Pomerleau and Todd Jochem

Phone: 412-268-3210 Fax: 412-268-5570

The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

and

AssistWare Technology Inc., 2498 Elkridge Drive, Wexford, PA 15090, USA

1. Introduction

Nearly 15,000 people die each year in the US in single vehicle roadway departure crashes [10]. These accidents are often caused by driver inattention, or driver impairment (e.g. fatigued or intoxicated drivers). A system capable of warning the driver when the vehicle starts to depart the roadway, or controlling the lateral position of the vehicle to keep it in its lane, could potentially eliminate many of these crashes. Nearly 70% of these crashes occur in rural or suburban settings on undivided two lane roads [10]. Since it is unlikely these roads will be upgraded in the foreseeable future, a system for preventing these crashes must rely on the existing road structure.

Research into such systems has focused on machine vision techniques that detect particular features in video images of the road ahead of the vehicle, and determine the desired vehicle trajectory based on the relative positions of these features. Many of these systems [2] [4] [5] [7] rely on tracking specific features, such as lane markings, from one image to the next. Others depend on detecting regions of the image representing the road based on features such as color [1] [6] or texture [11].

All these systems have a common characteristic. They all have a strong, a priori model of the road's appearance, and employ hand programmed detection algorithms to locate these characteristic features. Unfortunately, roads are not always cooperative. Road markings vary dramatically depending on the type of road (e.g. suburban street vs. interstate highway), and the state or country in which it is located. For example, many California freeways use regularly spaced reflectors embedded in the roadway, not painted markings, to delineate lane boundaries. Further challenges result from the fact that the environmental context can greatly impact road appearance. Changes in illumination due to shadows, glare or darkness, and obstructions by other vehicles, rain, snow, salt or other foreign objects often cause dramatic changes in the road's appearance. Together these variations often invalidate the assumptions underlying vision algorithms, resulting in poor road detection performance.

Alternative approaches that combine machine vision and machine learning techniques have demonstrated an enhanced ability to cope with variations in road appearance [4] [8] [9]. ALVINN is a typical system of this type. ALVINN employs an artificial neural network to learn the characteristic features of particular roads under specific conditions. It utilizes this learned road model to determine how the vehicle should be steered in order to remain in its lane. While systems of this type have been quite successful at driving on a wide variety of road types under many different conditions, they have several shortcomings. First, the process of adapting to a new road requires a relatively extended "retraining" period, lasting at least several minutes. While this adaptation process is relative quick by machine learning standards, it is unquestionably too long in a domain like autonomous driving, where the vehicle may be travelling at nearly 30 meters per second. Second, the retraining process invariably requires human intervention in one form or another. These systems employ a supervised learning technique such a backpropagation, requiring the driver to physically demonstrate the correct steering behavior for the system to learn.

A truly flexible system should 1) flexibly exploit whatever features are available to determine vehicle location, 2) adapt almost instantly when the

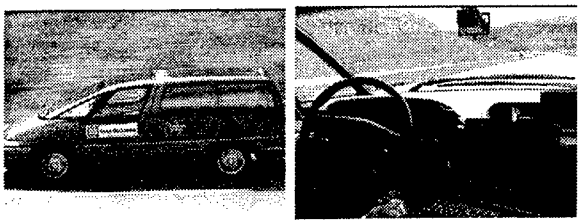


Figure 1: Exterior and interior views of the Navlab 5 testbed vehicle.

available features change, and 3) perform this adaptation without human supervision.

RALPH (Rapidly Adapting Lateral Position Handler) is a vision system developed jointly by Carnegie Mellon University and AssistWare Technology, Inc. which demonstrates these characteristics. RALPH decomposes the problem of steering a vehicle into three steps, 1) sampling of the image, 2) determining the road curvature, and 3) determining the lateral offset of the vehicle relative to the lane center. The output of the later two steps are combined into a steering command, which can be sent to the steering motor on our Navlab 5 testbed vehicle, shown in Figure 1, for autonomous steering control [3] or compared with the human driver's steering direction as part of a road departure warning system.

2. RALPH Sensor Configuration

A typical scene of the road ahead, as imaged by a video camera mounted next to the rearview mirror on Navlab 5, is depicted on the left of Figure 2. RALPH can utilize either black and white or color images, using a color-based contrast enhancement technique described in [8]. Obviously much of this image is irrelevant for the driving task (e.g. the parts of the image depicting the sky or the dashboard of the vehicle). These parts of the scene are eliminated, and only the portions of the scene inside the red trapezoid are processed. While the lower and upper boundaries of this trapezoid vary with vehicle velocity (moving further ahead of the vehicle, towards the top of the image, as vehicle speed increases), they typically project to approximately 20m and 70m ahead of the vehicle, respectively.

The second, and perhaps more important aspect of the trapezoid's shape is its horizontal extent. It is configured so that its width on the ground plane is identical at each row of the image. The horizontal distance that each row of the trapezoid encom-

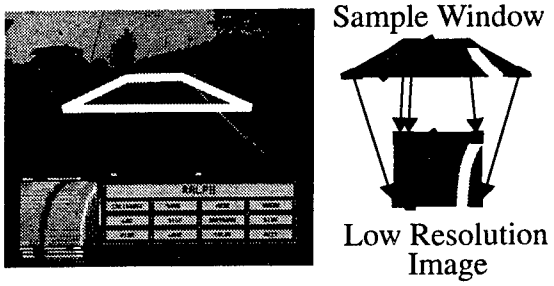


Figure 2: Forward looking image (left), and RALPH's sampling strategy (right).

passes is approximately 7.0 meters, about twice the width of a typical lane. This trapezoid is selectively sampled according to the strategy depicted in the schematic on the right of Figure 2. This sampling process creates a low resolution (30x32 pixel) image in which important features such as lane markings, which converged towards the top of the original image, now appear parallel in the low resolution image. Note that this image resampling is a simple geometric transformation, and requires no explicit feature detection.

2.1. Curvature Calculation

The "parallelization" of road features described above is crucial for the second step of RALPH processing, curvature determination. To determine the curvature of the road ahead, RALPH utilizes an "hypothesize and test" strategy. RALPH hypothesizes a possible curvature for the road ahead, subtracts this curvature from the parallelized low resolution image, and tests to see how well the hypothesized curvature has "straightened" the image.

The process RALPH utilizes to determine curvature is depicted in Figure 3. In this example, five curvatures are hypothesized for the original image, shown at the top. For each of the five hypothesized curvatures, the rows of the image are differentially shifted in an attempt to "undo" the curve and straighten out the image features. For left curve hypotheses, rows are shifted towards the right and for right curve hypotheses, rows are shifted left. For the more extreme hypothesized curvatures (on the far left and right), the rows of the original image are shifted further than for the less extreme curvatures (in the middle). For all the hypothesized curvatures, rows near the top of the image, corresponding to regions on the ground plane further

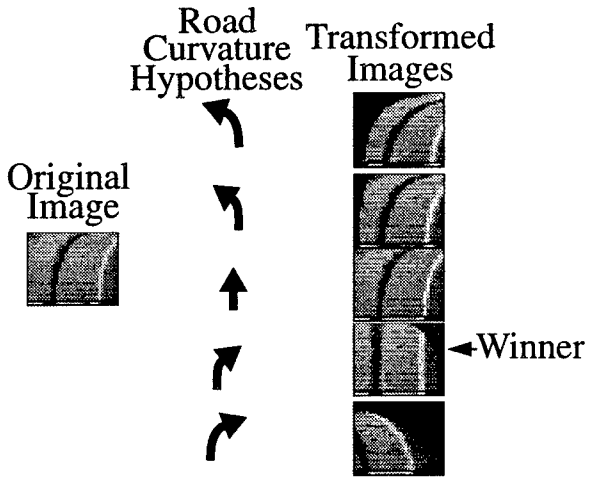


Figure 3: RALPH curvature hypotheses

ahead of the vehicle, are shifted further horizontally than rows near the bottom of the image. This differential shifting accounts for the fact that for a given hypothesized curvature, the road will be displaced more at the top of the image, far ahead of the vehicle, than at the bottom. The exact shift distance for each row in the transformed images is determined both by the geometry of the camera and the particular curvature hypothesis being tested.

As can be seen from Figure 3, the second curvature hypothesis from the right, corresponding to a shallow right turn, has resulted in a transformed image with the straightest features, and therefore should be considered the winning hypothesis. The technique used to score the "straightness" of each hypothesis is depicted in Figure 4. After differentially shifting the rows of the image according to a particular hypothesis, columns of the resulting transformed image are summed vertically to create a scanline intensity profile, shown in the two curves at the bottom of Figure 4. When the visible image features have been straightened correctly, there will be sharp discontinuities between adjacent columns in the image, as show in the right scanline intensity profile in Figure 4. In contrast, when the hypothesized curvature has shifted the image features too much or too little, there will be smooth transitions between adjacent columns of the scanline intensity profile, as depicted in the left profile of Figure 4. By summing the maximum absolute differences between intensities of adjacent columns in the scanline intensity profile, this property can be quantified to determine the curvature hypothesis that best straightens the image features.

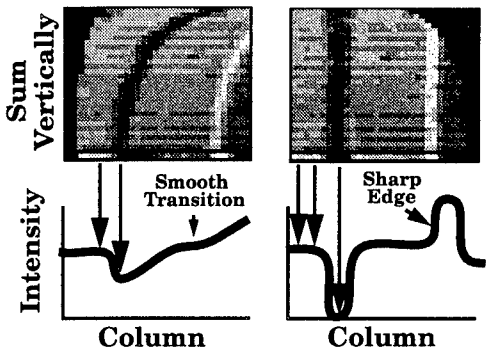


Figure 4: RALPH curvature scoring technique.

An important attribute to note about this technique for determining road curvature is that it is entirely independent of the particular features present in the image. As long as there are visible features running parallel to the road, this technique will exploit them to determine road curvature. These features need not be located at any particular position relative to the road, and need not have distinct boundaries - characteristics required by systems that utilize strong a priori road models and edge detection.

2.2. Lateral Offset Calculation

The next step in RALPH's processing is to determine the vehicle's lateral position relative to the lane center. This is accomplished using a template matching approach on the scanline intensity profile generated in the curvature estimation step. The scanline intensity profile is a one dimensional representation of the road's appearance as seen from the vehicle's current lateral position. By comparing this current appearance with the appearance of a template created when the vehicle was centered in the lane, the vehicle's current lateral offset can be estimated.

Figure 5 illustrates this lateral offset estimation procedure in more detail. Here, the current scanline intensity profile is depicted on the left, and the template scanline intensity profile, generated when the vehicle was centered in the lane, is depicted on the right. By iteratively shifting the current scanline intensity profile to the left and right, the system can determine the shift required to maximize the match between the two profiles (as measured by the corre-

lation between the two curves). The shift distance required to achieve the best match is proportional to the vehicle's current lateral offset.

Note that as with the curvature determination step, this process does not require any particular features be present in the image. As long as the visible features produce a distinct scanline intensity profile, the correlation based matching procedure can determine the vehicle's lateral offset. In particular, even features without distinct edges, such as pavement discoloration due to tire wear or oil spots, generate identifiable scanline intensity profile variations which RALPH can exploit to determine lateral offset. This is a performance feature which edge-based road detection systems do not share.

2.3. Adaptation to Changing Conditions

Another important feature of RALPH stems from the simplicity of its scanline intensity profile representation of road appearance. The 32 element template scanline intensity profile vector is all that needs to be modified to allow RALPH to handle a new road type. Modifying this vector is extremely easy. In the current RALPH implementation there are four ways of adapting the template to changing conditions.

The first method involves the driver centering the vehicle in its lane, and pressing a button to indicate that RALPH should create a new template. In under 100 msec, RALPH performs the processing steps described above to create a scanline intensity profile for the current road, and then saves it as the template. From that point on, RALPH can either actively control the steering wheel or warn the driver of road departure danger, using the newly created template to determine the vehicle's position relative to the lane center.

A second method for acquiring a template appropriate for the current road type is to select one from a library of stored templates recorded previously on a variety of roads. RALPH can select the best template for the current conditions by testing several of these previously recorded templates to determine which has the highest correlation with the scanline intensity profile created for the current image.

The third method of template modification occurs after an appropriate template has been selected. During operation, RALPH slowly evolves the current template by adding a small percentage of the current scanline intensity profile to the template. This allows the current template to adapt to gradual changes in the road's appearance, such as those caused by changes in the sun's angle.

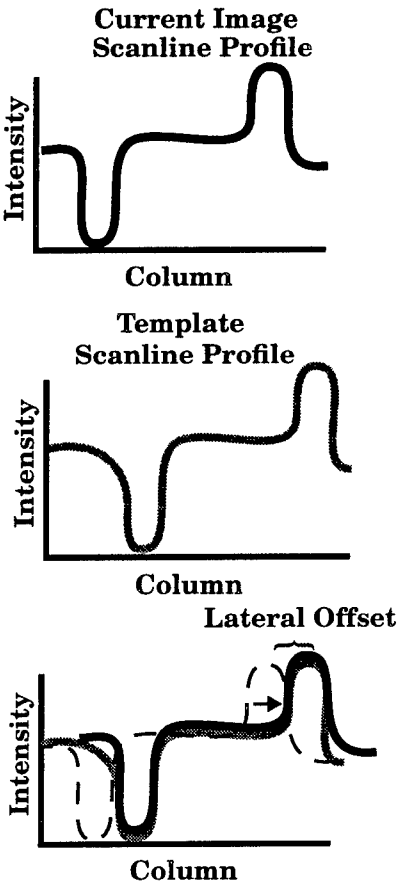


Figure 5: RALPH lateral offset determination technique.

RALPH handles more abrupt scene changes, such as changes in lane marker configuration, using the final and most interesting template modification strategy. In this technique, RALPH uses the appearance of the road in the foreground to determine the vehicle's current lateral offset and the curvature of the road ahead, as described above. At the same time, RALPH is constantly creating a new "rapidly adapting template" based on the appearance of the road far ahead of the vehicle (typically 70-100 meters ahead). This rapidly adapting template is created by processing the distant rows of the image in the same manner as described previously. The roads curvature is assumed to be nearly constant between the foreground and background, allowing RALPH to determine where the road is

ahead and therefore what the new template should look like when the vehicle is centered in its lane.

If the appearance of the road ahead changes dramatically, RALPH uses this technique to quickly create a template appropriate for the new road appearance. When the vehicle actually reaches the new road, RALPH determines that the template it was previously using is no longer appropriate, since it does not match the scanline intensity profile of the current image. It therefore swaps in the rapidly adapting template, and continues driving. Note that this rapid adaptation occurs in the time span of approximately 2 seconds, without any human intervention.

3. RALPH Performance

Extensive laboratory, test track and on-road experiments with the RALPH system have been conducted in order to characterize its performance. The results of these tests, presented below, indicate that RALPH can accurately estimate the vehicle's lateral position on the road, as well as the curvature of the road ahead, under a wide variety of conditions.

3.1. Laboratory Tests

An important factor determining autonomous driving effectiveness is the accuracy of the sensing system employed. The crucial accuracy metric for RALPH is how well can it estimate the location of the road ahead of the vehicle, since it is the road location that will be used to determine the direction to steer the vehicle.

In order to quantify RALPH's ability to accurately determine the position of the road ahead, controlled laboratory tests were conducted in which accurate measurements of the road's actual location could be made. To facilitate these measurements, high quality video sequences of road scenes were collected using a Umatic 3/4 inch VCR. These scenes were gathered in the Navlab 5 test vehicle, using the same camera mounted in the same location (next to the rear view mirror) as in the on-vehicle experiments described in following sections. These sequences include both day and night operation, as well as images of a variety of road types, including both rural roads and multi-lane divided highways. The test road sequences recorded on videotape were all between four and nine miles in length. While recording the sequences, the driver repeatedly changed the vehicle's lateral position within the lane in order to obtain a wide range of images.

The video sequences were subsequently replayed in the laboratory, and RALPH was used to track the road. More specifically, RALPH combined its estimates of the vehicle's lateral offset and the curvature of the road ahead into an estimate of the lane center location one second ahead (about 25m) of the vehicle.

RALPH's lane center position estimate was compared in real time with the estimate of lane center provided manually by the experimenter. The experimenter continuously indicated his estimate of the lane center location by keeping a crosshair centered over the right lane marking one second ahead of the vehicle in the image using a computer mouse. The difference between RALPH's estimate of lane position and the experimenter's estimate was stored for later analysis.

The results of these tests are summarized in Table Table 1. For each of the conditions tested, the table shows the mean and standard deviation of the difference between RALPH's estimate of the lane center position, and the experimenter's estimate of the lane center position. In general, RALPH's performance was quite good in all the conditions tested, with a total mean disagreement between RALPH and the experimenter of 13.2cm, which is just slightly larger than the width of a typical lane edge marker. As was expected, lower mean and standard deviation was observed in the conditions with the most consistent features. One such situation is shown in Figure 6. It depicts a daytime highway scene in which the lane markers are very clearly visible. Under these conditions, the mean disagreement between RALPH and the experimenter was 11.4cm. The variance of the disagreement was 14.3cm. Note that a substantial portion of the disagreement between RALPH and the experimenter can be attributed to inconsistency in the experimenter's estimate of the lane center position. Accurately indicating the lane position 20m ahead using a mouse is a difficult task. In a series of repeatability tests, it was determined that the experimenter's estimate of lane position over two different trials on the same section of videotape varied by an average of 7.3cm.

Table 1: RALPH lane location estimation accuracy

Condition	Mean Error (cm)	Error Std. Dev. (cm)
Daytime Highway	11.4	14.3
Daytime Highway w/ Shadows	13.8	18.9
Nighttime Highway	11.1	13.8
Daytime Rural Road	13.7	16.2
Daytime Rural Road w/ Glare	15.8	17.2
Nighttime Rural Road	13.8	16.8
Total	13.2	16.2

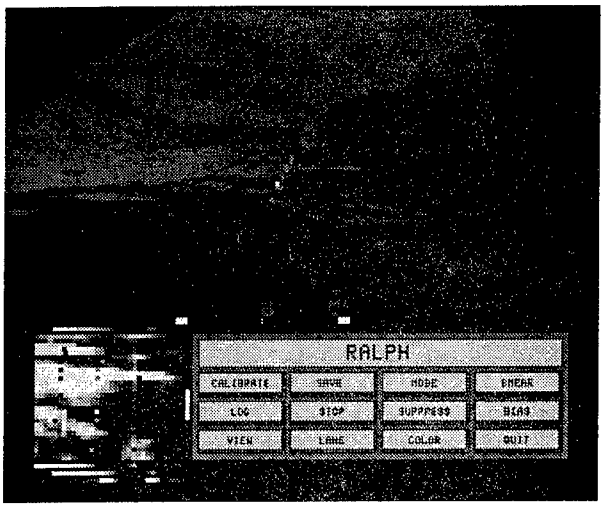


Figure 7: RALPH processing a daytime highway image with heavy shadows.

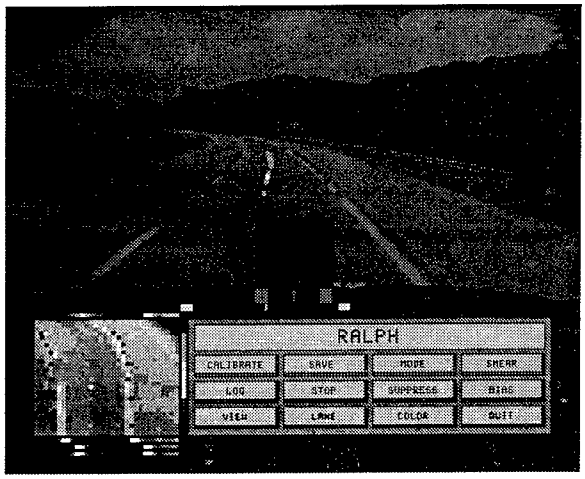


Figure 6: RALPH processing a daytime highway image.

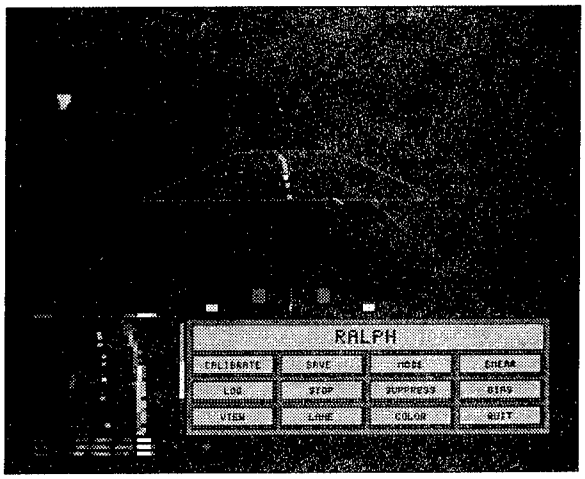


Figure 8: RALPH processing a nighttime highway image.

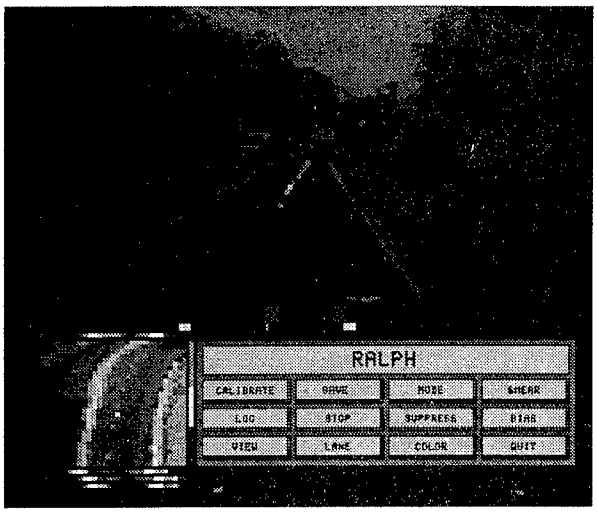


Figure 9: RALPH processing a daytime rural road image.

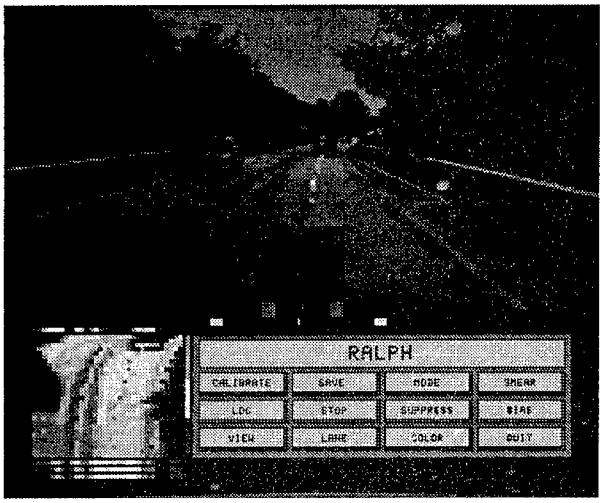


Figure 10: RALPH processing an early morning rural road image with glare off road.

On the same stretch of highway under conditions of heavy shadows (See Figure 7), the mean and standard deviation of RALPH's lane position estimation error increased somewhat to 13.8cm and 18.9cm, respectively. This increase in error was due primarily to the limited dynamic range of the camera, causing the shadowed regions of the image to be black and/or the areas in sunlight to be saturated.

In contrast, RALPH's lane location on the same stretch of highway ability improved slightly at night. As can be seen in Figure 8, the lane markers were very distinct in this situation, resulting in a mean error of 11.1cm and a standard deviation of 13.8cm.

RALPH's performance on rural roads such as the one in Figures 9 was fairly similar to the highway results. The mean and standard deviation under favorable daytime conditions did increase slightly over the corresponding figures for favorable daytime highway images, to 13.7cm and 16.2cm, respectively. This increase was primarily caused by two factors. First, hills on the rural roads changed the perspective of the camera relative to the road. This resulted in slight additional lane position estimation errors, particularly at grade transition points. Second, there were several cross streets intersecting the section of rural road tested, which occasionally resulted in momentary inaccuracy when the lane markers disappeared. However the increase in average lane position estimation error due to these effects was small, on the order of two centimeters.

One problem with lane tracking systems which rely exclusively on lane markers to locate the road ahead is that they sometimes have difficulty when glare off the pavement makes the markers hard to find. This type of glare typically occurs when the pavement is wet, and/or when the sun is low on the horizon. To quantify the effect of these conditions on RALPH, a video sequence was collected on the same rural road during the early morning hours heading into the rising sun. An example image from this sequence is shown in Figure 10. As was expected, the mean and standard deviation of RALPH's error increased under these conditions, to 15.8cm and 17.2cm, respectively. However these increases were slight, again in the range of 2cm. RALPH was still able to accurately locate the road ahead under these conditions by adapting its processing to utilize the boundary between the bright pavement and the dark shoulder. This ability to adapt to changing conditions was determined to be particularly important in the on-road tests, described in Section 3.3.

In summary, the laboratory tests indicate that RALPH can localize the position of the road ahead of the vehicle to within approximately the width of a single lane marker under a variety of conditions. To further characterize RALPH's ability to perform repeatably and reliably, we also performed extensive test track and on-road experiments.

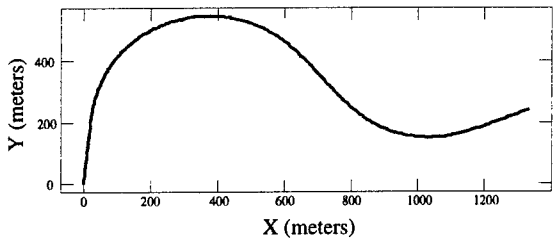


Figure 11: S-curve used for testing RALPH.

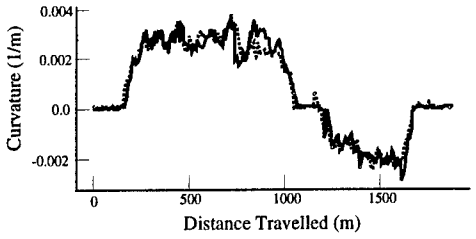


Figure 12: RALPH's curvature estimate on two traversals through the s-curve.

3.2. Test Track Experiments

Additional controlled experiments were conducted on a road segment outside of Pittsburgh often used for testing. These tests involved repeatedly driving the same stretch of roadway at different speeds when there were no other vehicles on the test road.

In the videotape experiments presented above, the goal was to quantify RALPH's ability to find the position of the road ahead by combining RALPH's estimate of the vehicle's lateral position and its estimate of the curvature of the road ahead. In the first set of test track experiments, the goal was to tease apart this combination, and measure RALPH's ability estimate the curvature of the road ahead. In this experiment, the Navlab 5 test vehicle was driven manually through the S-curve shown in Figure 11.

Careful measurement of the first curve indicates that it has an average radius of curvature of approximately 343m. Figure 12 shows RALPH's estimate of the road curvature during two traversals of the entire S-curve at 55mph.

Note the consistency in the curvature estimate between the two traversals. RALPH's mean estimate for radius of curvature during the first traversal of the first curve was 373m, and the mean on the second traversal was 374m. Not only are the two estimates extremely close, but they match quite closely to the measured radius of 343m. In

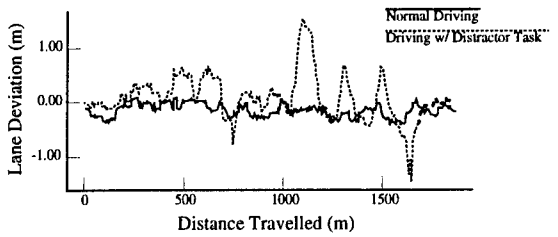


Figure 13: Lane deviation in normal driving, and when the driver is distracted.

fact, the 30m discrepancy between the measured curve radius and RALPH's estimate may at least partially be attributed to uncertainty in the manual curvature measurement.

The next set of experiments was done to determine if anomalous driver behavior can be detected using RALPH. This information is valuable not for autonomous driving, but rather for determining RALPH's applicability as a tool to prevent roadway departure crashes. Again the driver drove twice through the S-curve at 55mph. The first time through, the driver concentrated on accurate driving. The second time through, the driver was momentarily distracted by an in-cab distractor task. The distractor task required the driver to glance to the back of the vehicle for up to two seconds. The goal was to determine if the lane deviations resulting from this momentary inattention could be detected in the lane tracking output RALPH produces.

A graph of RALPH's estimate of the vehicle's lateral position, both during normal driving and while the driver was performing the distractor task are shown in Figure 13. As can be seen from the graph, the relatively large magnitude lane deviations resulting from momentary distraction are clearly discernible when compared with driver's normal lane deviations. This characteristic is extremely useful when using RALPH as a roadway departure warning system.

The results of these test track experiments indicate that RALPH can repeatably detect both the curvature of the road ahead, as well as the excessive lane deviation by the driver. However these experiments neglected two important aspects of the autonomous driving task. First, in the test track experiments described above, RALPH was passively monitoring the vehicle's position on the roadway and the curvature of the road ahead. RALPH's ability to Combine these measurements into a command for

the steering wheel which will keep the vehicle centered in its lane was not tested. In addition, these experiments were conducted under favorable weather and lighting conditions.

3.3. Open Road Tests

One of the most significant potential drawbacks of driving systems that rely on video cameras for sensor input is their susceptibility to adverse conditions. Systems that rely on visible features to determine the vehicle's position on the road can have trouble when these distinctions become difficult to detect, due to adverse weather, poor lighting, or degraded pavement. To quantify this effect, a series of on-road tests of the RALPH system was conducted.

The culmination of these experiments was a 2850 mile test drive from Washington, DC to San Diego, CA in which RALPH's steering commands were used to control the Navlab 5 testbed vehicle. Except for a few detours, the trip exclusively involved highway driving. The trip included many of the difficulties typically encountered in normal driving - nighttime driving, driving at sunset when the sun is low on the horizon, driving through rain storms, driving on poorly marked roads, and driving through construction areas.

During the 2850 mile trip, statistics about the RALPH's driving performance were collected. The primary metric was the percent of the total trip distance that RALPH was controlling the steering wheel. To measure this value, the assumption was made that if the steering wheel position disagreed significantly from RALPH's commanded position, the safety driver had taken control of the wheel[1]. In more detail, when the steering direction suggested by RALPH differed from the actual steering wheel position such that following RALPH's steering arc at the current speed would result in a difference in lateral acceleration of 0.04g or greater, then RALPH was judged to have been overridden by the safety driver.

Overall, the results were quite encouraging. Using the above metric, RALPH was able to steer the vehicle autonomously for 98.1 percent (2796/2850 miles) of the trip. Due to the system's ability to adapt to changing conditions, RALPH was able to drive in situations which would be difficult for other lane keeping systems, particularly those that rely on finding distinct lane markers. Some of the different situations that RALPH was able to handle are illustrated in Figures 14 through 18.

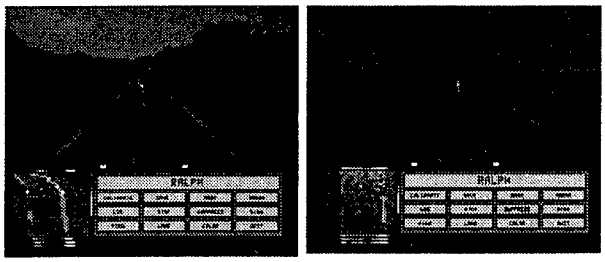


Figure 14: Examples of well marked roadway encountered in cross country test.

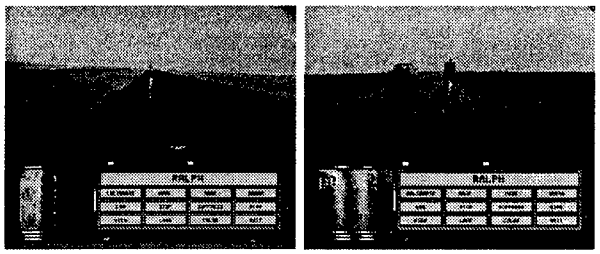


Figure 15: Roads without strong markings (left) and with wet pavement (right).

Some of the roads, like the two shown in Figure 14, were very much like one would expect on a major highway - nice pavement and good lane markings. Even when the lane markers we missing, as on the freshly paved road in the left hand image of Figure 15, RALPH was able to continue driving by exploiting the boundary between the pavement and the off road area. This same type of road proved quite difficult at night however, when the edge formed by the pavement boundary was no longer visible. In particular, on the third night of the trip a ten mile stretch of new, unpainted highway like the one shown in the left image of Figure 15 accounted for a significant portion of the 1.9 percent of the distance that RALPH was not able to drive. Rain proved to be less of a problem. Even when the specular reflection off wet pavement obscured the lane markings, as in the right hand image of Figure 15, RALPH was able to key off other, more subtle variations in the road's appearance to determine how it should steer. These additional features were typically formed by water pooling in ruts on the

1. The motor on the Navlab 5 steering wheel used by RALPH to control the vehicle is purposefully weak, allowing the safety driver to easily override the command steering direction when necessary by simply overpowering the motor.

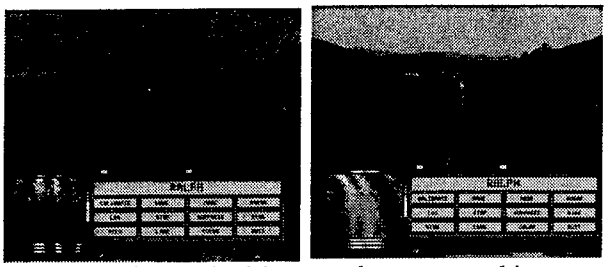


Figure 16: Road with severely worn markings (left) and unpaved road (right).

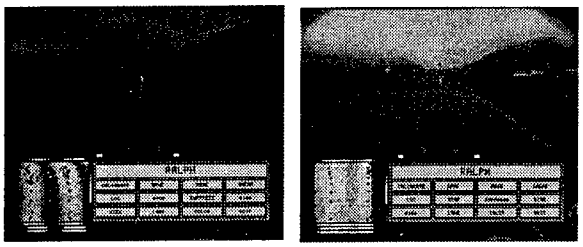


Figure 17: California freeways with reflectors instead of painted lane markings.

road, and by the tires of vehicle's ahead leaving tracks in the wet pavement.

West of the Rocky Mountains, there were stretches of very poor roads, two of which are shown in Figure 16. Often the lane markers were nearly invisible due to wear (left). Several times there were long stretches of construction where the road was composed of a very fine, packed gravel, without any lane markings (right). During these stretches, RALPH was able to exploit the differences in appearance of the packed and loose gravel to continue driving.

The freeways in California posed an interesting challenge. Instead of having painted markings to delineate lanes, they have reflectors that are nearly invisible during the day (See left image, Figure 17). In these situations, RALPH was able to drive using the diffuse discoloration from the oil spot down the center of the lane. RALPH also performed well on the I-15 HOV lane into San Diego, which have no visible lane markings, but a strong boundary between the cement road surface and the asphalt shoulder (right image, Figure 17).

The situation which gave the system the most difficulty was in city traffic, when the road markings were either missing or obscured by other traffic (See Figure 18). But in this and most of the other situations RALPH had difficulty with, it was able to recognize that it could not correctly steer, and inform the safety driver of its confusion.

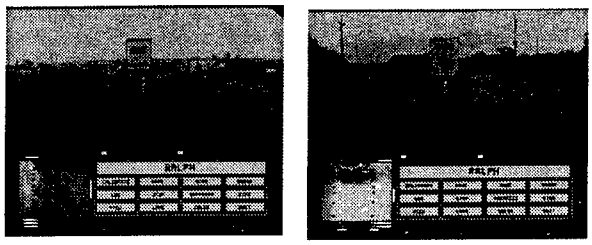


Figure 18: Challenging images from city driving.

4. Conclusion and Future Work

In conclusion, extensive tests of the RALPH vision-based lane position estimation system indicate that it is able to accurately detect the vehicle's position and orientation relative to the roadway in a wide variety of situations and can use this information to steer our tesbed vehicle. Current work on RALPH focuses on minimizing those few remaining conditions that do provide difficulty for RALPH, using technique such as active camera control to focus the system's attention on important aspects of the scene. In addition, work is currently under way to develop techniques which allow RALPH to reliably determine error bounds on its estimate of the road's location ahead.

The simplicity of the RALPH algorithm suggests that a custom hardware implementation should be feasible. This has the potential to dramatically reduce both the size and the cost of subsequent versions of RALPH. Our eventual goal is to build a system that is small enough to fit behind the rearview mirror, and inexpensive enough to sell as option on passenger cars. Initially such a system would simply warn the driver if he is drifting off the road. In time such a system could potentially assume at least partial control, relieving the driver of the monotonous task of steering, just as standard cruise control has done for maintaining vehicle speed.

5. References

[1] Crisman, J. and Thorpe, C. (1990) "Color vision for road following". *Vision and Navigation: The CMU Navlab*. C. Thorpe (ed.) Kluwer Academic Publishing, Boston MA.

[2] Dickmanns, E. D., Behringer, R., Dickmanns D., Hildebrandt, T., Maurer, M., Thomanek, F., and Schielen, J., "The seeing passenger car 'VaMoRs-P,'" *1994*

IEEE Symposium on Intelligent Vehicles, pp. 68-73.

[3] Jochem, T., Pomerleau, D., Kumar, B. and Armstrong, J. (1995) "PANS: A portable navigation platform". *1995 IEEE Symposium on Intelligent Vehicles.*

[4] Kim, K.I., Oh, S.Y., Lee, J.S., Han, J.H., and Lee, C.N. (1993) "An autonomous land vehicle: design concept and preliminary road test results". *1993 IEEE Symposium on Intelligent Vehicles*, pp. 146-151.

[5] Kluge, K. and Thorpe, C. (1992) "Representation and recovery of road geometry in YARF", *1992 IEEE Symposium on Intelligent Vehicles*, pp. 114-119.

[6] Marra, M., Dunlay, T.R. and Mathis, D. (1988) "Terrain classification using texture for the ALV." Martin Marietta Information and Communications Systems technical report 1007-10.

[7] Nashman, M. and Schneiderman, H. (1993) "Real-time visual processing for autonomous driving". *1993 IEEE Symposium on Intelligent Vehicles*, pp. 373-378.

[8] Pomerleau, D. A. (1994) *Neural Network Perception for Mobile Robot Guidance*, Kluwer Academic Publishing, Boston MA.

[9] Rosenblum, M. and Davis, L.S. (1993) "The use of a radial basis function network for visual autonomous road following". *1993 IEEE Symposium on Intelligent Vehicles*, pp. 432-439.

[10] Want, J.S. and Knipling, R.R. (1993) *Single-Vehicle Roadway Departure Crashes: Problem Size Assessment and Statistical Description.* National Highway Traffic Safety Administration Technical Report DTNH-22-91-C-03121.

[11] Zhang, J. and Nagel, H. (to appear) "Texture analysis and model-based road recognition for autonomous driving". To appear in *Journal of Computer Vision, Graphics and Image Processing.*

Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques

Michael A. Smith
Dept. of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
msmith@cs.cmu.edu
http://www.cs.cmu.edu/~msmith

Takeo Kanade
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
tk@cs.cmu.edu
http://www.cs.cmu.edu/~tk

Abstract

Digital video is rapidly becoming important for education, entertainment, and a host of multimedia applications. With the size of the video collections growing to thousands of hours, technology is needed to effectively browse segments in a short time without losing the content of the video. We propose a method to extract the significant audio and video information and create a "skim" video which represents a very short synopsis of the original. The goal of this work is to show the utility of integrating language and image understanding techniques for video skimming by extraction of significant information, such as specific objects, audio keywords and relevant video structure. The resulting skim video is much shorter, where compaction is as high as 20:1, and yet retains the essential content of the original segment.

1 Introduction

With increased computing power and electronic storage capacity, the potential for large digital video libraries is growing rapidly. These libraries, such as the Informedia™ Project at Carnegie Mellon [Wactlar *et al.*, 1996], will make thousands of hours of video available to a user. For many users, the video of interest is not always a full-length film. Unlike video-on-demand, video libraries should provide informational access in the form of brief, content-specific segments as well as full-featured videos.

Even with intelligent content-based search algorithms being developed [Mauldin, 1989], [TREC, 1993], multiple video segments will be returned for a given query to insure retrieval of pertinent information. The users will often need to view all the segments to obtain their final selections. Instead, the user will want to "skim" the relevant portions of video for the segments related to their query.

1.1 Browsing Digital Video

Simplistic browsing techniques, such as fast-forward playback and skipping video frames at fixed intervals, reduce video viewing time. However, fast playback perturbs the audio and distorts much of the image information [Degen *et al.*, 1992], and displaying video sections at fixed intervals merely gives a random sample of the overall content. Another idea is to present a set of "representative" video frames (e.g. keyframes in motion-based encoding) simultaneously on a display screen. While useful and effective, such static displays miss an important aspect of video: video contains audio information. It is critical to use and present audio information, as well as image information, for browsing. Recently, researchers have proposed browsing representations based on information within the video [Zhang *et al.*, 1993], [Arman *et al.*, 1994b]. These systems rely on the motion in a scene, placement of scene breaks, or image statistics, such as color and shape, but they do not make integrated use of image and language understanding.

*This work was sponsored by the National Science Foundation under grant no. IRI- 9411299, the National Space and Aeronautics Administration, and the Advanced Research Projects Agency. Michael Smith is sponsored by Bell Laboratories. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the United States Government or Bell Laboratories.

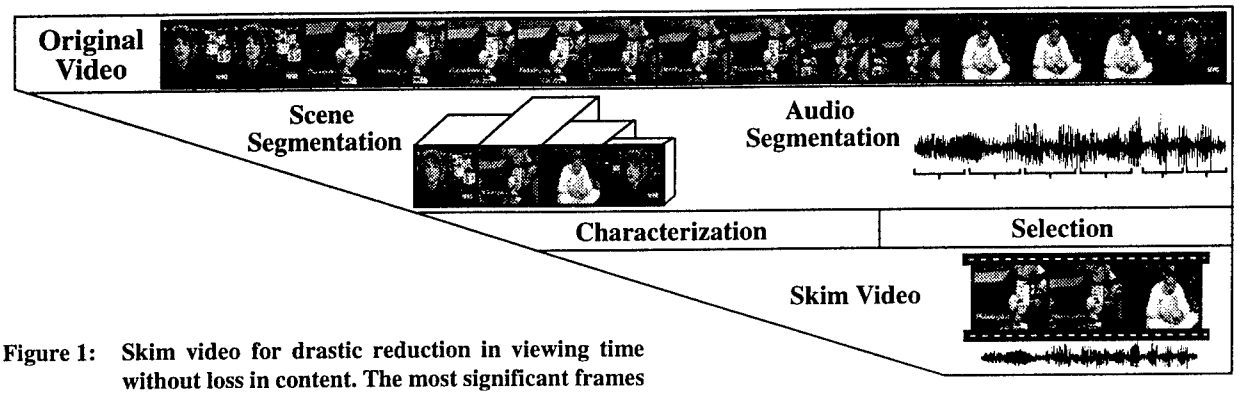


Figure 1: Skim video for drastic reduction in viewing time without loss in content. The most significant frames from a selected scene are chosen for browsing.

An ideal browser would display only the video pertaining to a segment's content, suppressing irrelevant data. It would show less video than the original and could be used to sample many segments without viewing each in its entirety. The amount of content displayed should be adjustable so the user can view as much or as little video as needed, from extremely compact to full-length video. The audio portion of this video should also consist of the significant audio or spoken words, instead of simply using the synchronized portion corresponding to the selected video frames.

1.2 Video Skims

Figure 1 illustrates the concept of extracting the most representative video frames and audio information to create the skim. The critical aspect of compacting a video is context understanding, which is the key to choosing the "significant images and words" that should be included in the skim video. We characterize the significance of video through the integration of image and language understanding. Segment breaks produced by image processing can be examined along with boundaries of topics identified by the language processing of the transcript. The relative importance of each scene can be evaluated by 1) the objects that appear in it, 2) the associated words, and 3) the structure of the video scene. The integration of language and image understanding is needed to realize this level of characterization and is essential to skim creation.

In the sections that follow, we describe the technology involved in video characterization from audio and images embedded within the video, and the process of integrating this information for skim creation.

2 Video Characterization

Through techniques in image and language understanding, we can characterize scenes, segments, and individual frames in video. Figure 2 illustrates characterization of a segment taken from a video titled "Destruction of Species", from WQED Pittsburgh. At the moment, language understanding entails identifying the most significant words in a given scene, and for image understanding, it entails segmentation of video into scenes, detection of objects of importance (face and text) and identification of the structual motion of a scene.

2.1 Language Characterization

Language analysis works on the transcript to identify important audio regions known as "keywords". We use the well-known technique of TF-IDF (Term

$$TF\text{-}IDF = \frac{f_s}{f_c} \qquad (1)$$

Frequency Inverse Document Frequency) to measure relative importance of words for the video document [Mauldin, 1989]. The TF-IDF of a word is its frequency in a given scene, f_s, divided by the frequency, f_c, of its appearance in a standard corpus. Words that appear often in a particular segment, but relatively infrequently in a standard corpus, receive the highest TF-IDF weights. A threshold is set to extract keywords from the TF-IDF weights, as shown in the bottom rows of Figure 2.

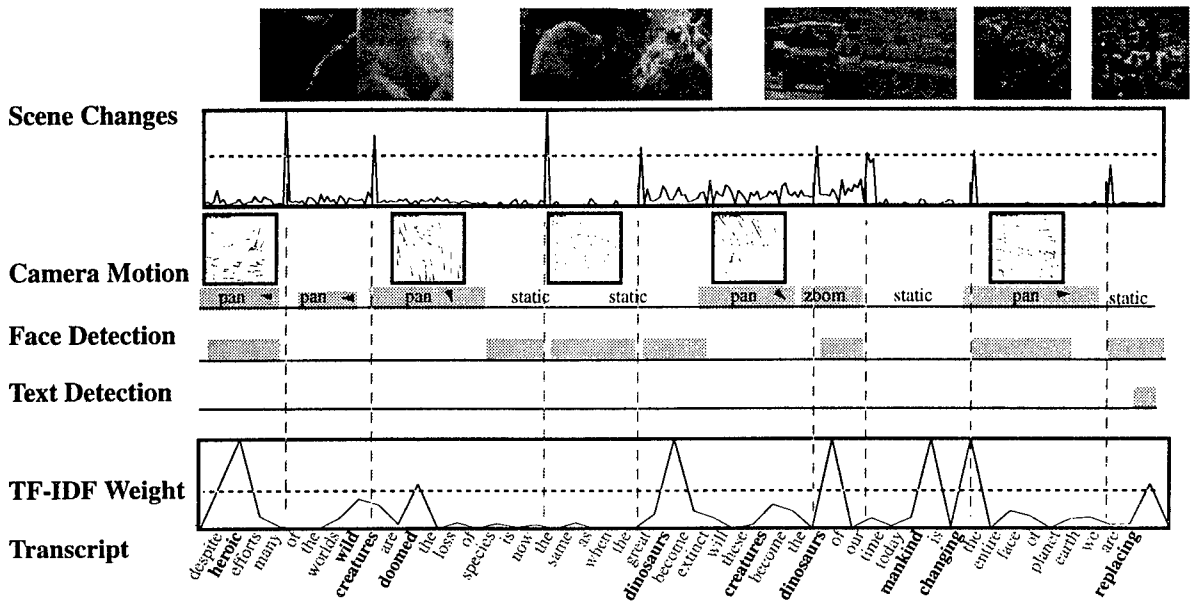
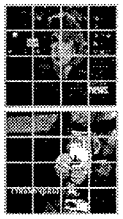


Figure 2: Video Characterization Technology. Video is segmented into scenes, and camera motion is detected along with significant objects (faces and text). Bars show frames with positive results.

2.2 Scene Segmentation

Many research groups have developed working techniques for detecting scene changes [Zhang et al., 1993], [Hampapur *et al.*, 1995], [Arman *et al.*, 1994a]. We choose to segment video by the use of a comparative color histogram difference measure. By detecting significant changes in the weighted



$$D(t) = \sum_{v=0}^{N} \left| H_t(v) - H_{t+1}(v) \right| \qquad (2)$$

$H_t(v)$: Histogram of Color in Image(t)

color histogram of each successive frame, video sequences are separated into scenes. Peaks in the difference, $D(t)$, are detected and an empirically set threshold is used to select scene breaks. We have found that this technique is simple, and yet robust enough to maintain high levels of accuracy for our purpose. Using this technique, we have achieved 91% accuracy in scene segmentation on a test set of roughly 495,000 images (5 hours). Examples of segmentation results are shown in the top row of Figure 2.

2.3 Camera Motion Analysis

One important aspect of video characterization is interpretation of camera motion. The global distribution of motion vectors distinguishes between object motion and actual camera motion. Object motion typically exhibits flow fields in specific regions of an image. Camera motion is characterized by flow throughout the entire image.

Motion vectors for each 16x16 block are available with little computation in the MPEG-1 video standard [MPEG-1, 1991]. An affine model is used

$$u(x_i, y_i) = ax_i + by_i + c \qquad (3)$$

$$v(x_i, y_i) = dx_i + ey_i + f \qquad (4)$$

to approximate the flow patterns consistent with all types of camera motion. Affine parameters $a,b,c,d,e,$ and f are calculated by minimizing the least squares error of the motion vectors. We also compute average flow $\bar{v}$ and $\bar{u}$.

Using the affine flow parameters and average flow, we classify the flow pattern. To determine if a pattern is a zoom, we first check if there is the convergence or divergence point (x_0,y_0), where $u(x_i, y_i) = 0$ and $v(x_i, y_i) = 0$. To solve for (x_0,y_0), the following relation must be true: $\begin{vmatrix} a & b \\ d & e \end{vmatrix} \neq 0$
If the above relation is true, and (x_0,y_0) is located inside the image, then it must represent the focus of expansion. If $\bar{v}$ and $\bar{u}$, are large, then this is the focus of the flow and camera is zooming. If (x_0,y_0) is outside the image, and $\bar{v}$ or $\bar{u}$ are large, then the

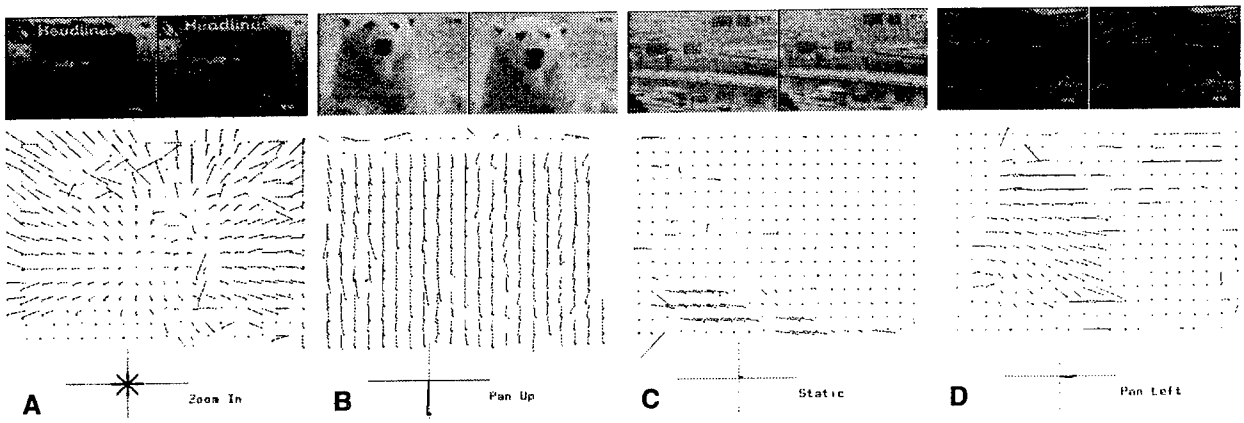


Figure 3: Camera motion analysis from MPEG motion vectors: A) Zoom distribution, B) Upward pan with subtle object motion, C) Static, D) Significant object motion detected as pan.

camera is panning in the direction of the dominant vector.

If the above determinant is approximately 0, then (x_0, y_0) does not exist and camera is panning or static. If $\bar{v}$ or $\bar{u}$ are large, the motion is panning in the direction of the dominant vector. Otherwise, there is no significant motion and the flow is static. We eliminate fragmented motion by averaging the results in a 20 frame window over time. Table 1 shows the statistics for detection on various sets of images. Regions detected are either pans or zooms. Examples of the camera motion analysis results are shown in Figure 3.

Table 1: Camera Motion Detection Results

Data(Images)	Regions Detected	Regions Missed	False Regions
Species I - II (20724)	23	5	1
PlanetEarthI-II (25680)	36	1	3
CNHAR News (30520)	14	1	2

Figure 4: Detection of human-faces.

2.4 Object Detection

Identifying significant objects that appear in the video frames is one of the key components for video characterization. For the time being, we have chosen to deal with two of the more interesting objects in video: human faces and text (caption characters). To reduce computation we detect text and faces every 15th frame.

2.4.1 Face Detection

The "talking head" image is common in interviews and news clips, and illustrates a clear example of video production focussing on an individual of interest. A human interacting within an environment is also a common theme in video. The human-face detection system used for our experiments was developed by Rowley, Baluja and Kanade [Rowley *et al.*, 1996]. It detects mostly frontal faces of any size and any background. Its current performance level is to detect over 86% of more than 507 faces contained in 130 images, while producing approximately 63 false detections. While improvement is needed, the system can detect faces of varying sizes and is especially reliable with frontal faces such as talking-head images. Figure 4 shows examples of its output, illustrating the range of face sizes that can be detected.

2.4.2 Text Detection

Text in the video provides significant information as to the content of a scene. For example, statistical numbers and titles are not usually spoken but are included in the captions for viewer inspection. A typical text region can be characterized as a hori-

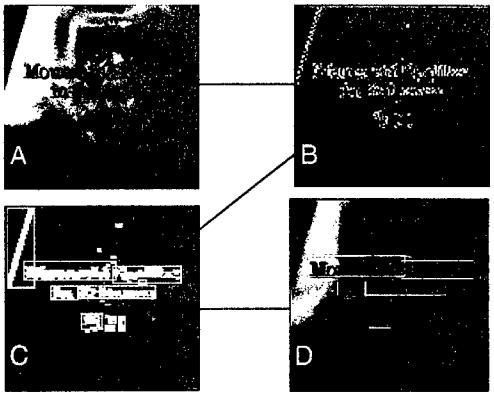
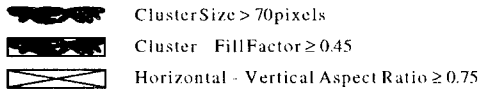


Figure 5: Stages of text detection: A) Input, B) Filtering, C) Clustering, and D) Region Extraction.

zontal rectangular structure of clustered sharp edges, because characters usually form regions of high contrast against the background. By detecting these properties we extract regions from video frames that contain textual information. Figure 5 illustrates the process of detecting text; primarily, regions of horizontal titles and captions.

We first apply a 3x3 horizontal differential filter to the entire image with appropriate binary thresholding for extraction of vertical edge features. Smoothing filters are then used to eliminate extraneous fragments, and to connect character sections that may have been detached. Individual regions are identified by cluster detection and their bounding rectangles are computed. Clusters with bounding regions that satisfy the following constraints are selected:



A cluster's bounding region must have a large horizontal-to-vertical aspect ratio as well as satisfying various limits in height and width. The fill factor of the region should be high to insure dense clusters. The cluster size should also be relatively large to avoid small fragments. An intensity histogram of each region is used to test for high contrast. This is because certain textures and shapes appear similar to text but exhibit low contrast when examined in a bounded region. Finally, consistent detection of the same region over a certain period of time is also tested since text regions are placed at the exact position for many video frames. Figure 6 shows detection examples of words and subsets of a word. Table 2 presents statistics for detection on various sets of images.

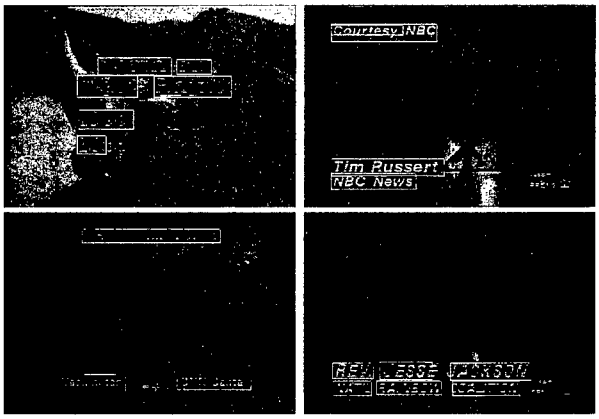


Figure 6: Text detection results with various images.

Table 2: Text Region Detection Results

Data (Images)	Regions Detected	Regions Missed	False Detections
CNHAV News (1056)	26	1	3
CNHAR News (1526)	48	0	5
Species I (264)	12	2	0
Planet Earth I-II(1712)	0	0	2

3 Technology Integration and Skim Creation

We have characterized video by scene breaks, camera motion, object appearance and keywords. Skim creation involves selecting the appropriate keywords and choosing a corresponding set of images. Candidates for the image portion of a skim are chosen by two types of rules: 1) Primitive Rules, independent rules that provide candidates for the selection of image regions for a given keyword, and 2) Meta-Rules, higher order rules that select a single candidate from the primitive rules according to global properties of the video. The subsections below describe the steps involved in the selection, prioritizing and ordering of the keywords and video frames.

3.1 Audio Skim

The first level of analysis for the skim is the creation of the reduced audio track, which is based on the keywords. Those words whose TF-IDF values are higher than a fixed threshold are selected as keywords. By varying this threshold, we control the number of keywords, and thus, the length of the skim. The length of the audio track is determined by a user specified compaction level.

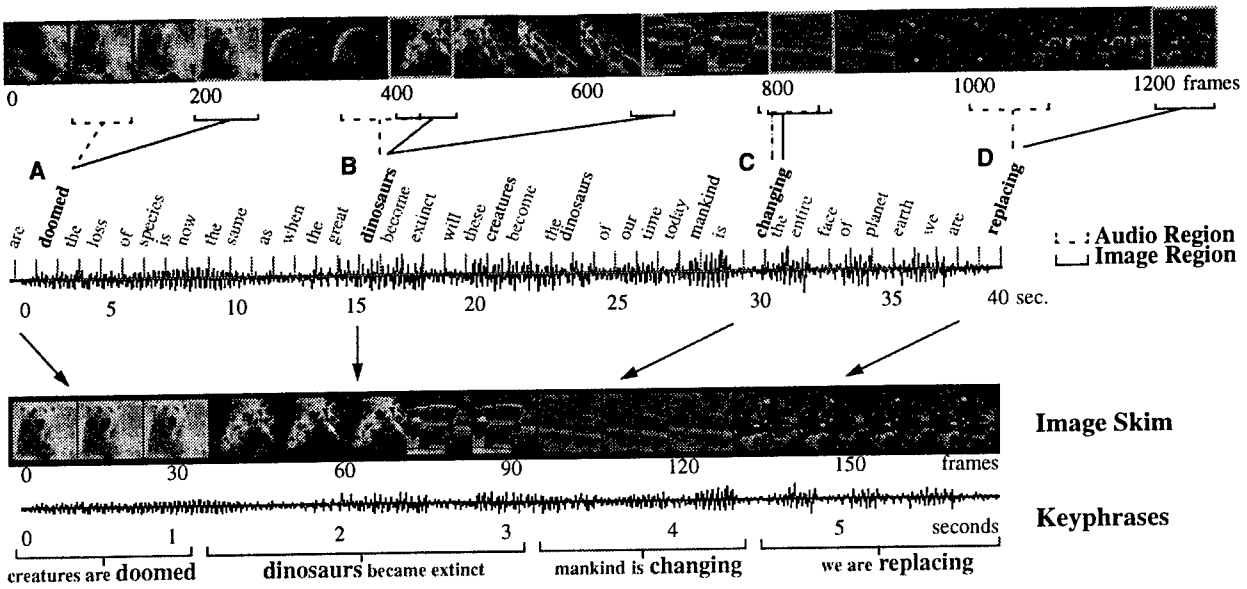


Figure 8: Skim creation incorporating word relevance, significant objects (humans and text), and camera motion: A) For the word "doomed", the images following the camera motion are selected, B) The keyphrase for "dinosaur" is long so portions of the next scene are used for more content, C) No significant structure for the word "changing", D) For the word "replacing" The latter portion of the scene contains both text and humans.

similar scenes that are less than 5 seconds apart, are used for skimming.

3. Short Sequences(SSN)
Short successive shots often introduce a more important topic. By measuring the duration of each scene, we can detect these regions and identify "short shot" sequences. The video frames that follow these sequences and the exact sequence are used for skimming.

4. Object Motion(OBM)
Object motion is import simply because video producers usually include this type of footage to show something in action. We are currently exploring ways to detect object motion in video.

5. Bounded Camera Motion(BCM/ZCM)
The video frames that preceed or follow a pan or zoom motion are usually the focus of the segment. We can isolate the video regions that are static and bounded by segments with motion, and therefore likely to be the focal point in a scene containing motion.

6. Human Faces and Captions(TXT/FAC)
A scene will often contain recognizable humans, as well as captioned text to describe the scene. If a scene contains both faces and text, the portion containing text is used for skimming. A lower level of priority is given to the scenes with video frames containing only human-faces or text. For these scenes priority is given to text.

7. Significant Audio(AUD)
If the audio is music, then the scene may not be used for skimming. Soft music is often used as a transitional tool, but seldom accompanies images of high importance. High audio levels (e.g. loud music, explosions) may imply an important scene is about to occur. The skim region will start after high audio levels or music.

8. Default Rule(DEF)
Default video frames align to audio keyphrases.

3.3 Image Adjustments

With prioritized video frames from each scene, we now have a suitable representation for combining the image and audio skims for the final skim. A set of higher order Meta-Rules are used to complete skim creation.

For visual clarity and comprehension, we allocate at least 30 video frames to a keyphrase. The 30 frame minimum for each scene is based on empirical studies of visual comprehension in short video sequences. When a keyphrase is longer than 60 video frames, we include frames from skim candidates of adjacent scenes within the 5 second search

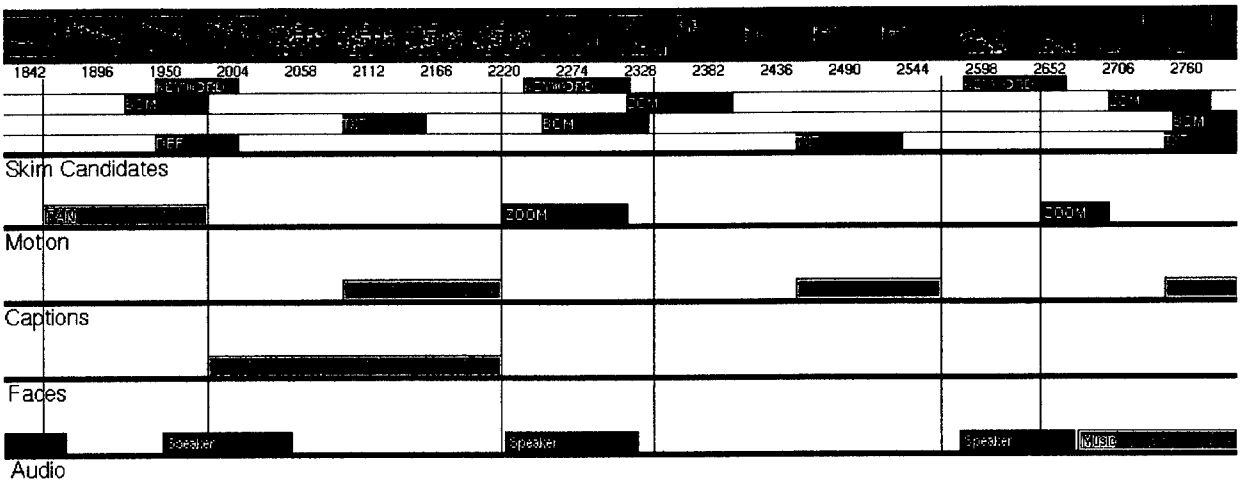


Figure 7: Characterization data with skim candidates and keyphrases for "Destruction of Species". The skim candidate symbols correspond to the following primitive rules: BCM, Bounded Camera Motion; ZCM, Zoom Camera Motion; TXT, Text Captions; and DEF Default. Vertical lines represent scene breaks.

Keywords that appear in close proximity or repeat throughout the transcript may create skims with redundant audio. Therefore, we discard keywords which repeat within a minimum number of frames (150 frames) and limit the repetition of each word.

Our experiments have shown that using individual keywords creates an audio skim which is fragmented and incomprehensible for some speakers. To increase comprehension, we use longer audio sequences, "keyphrases", in the audio skim. A keyphrase is obtained by starting with a keyword, and extending its boundaries to areas of silence or neighboring keywords. Each keyphrase is isolated from the original audio track to form the audio skim. The average keyphrase lasts 2 seconds.

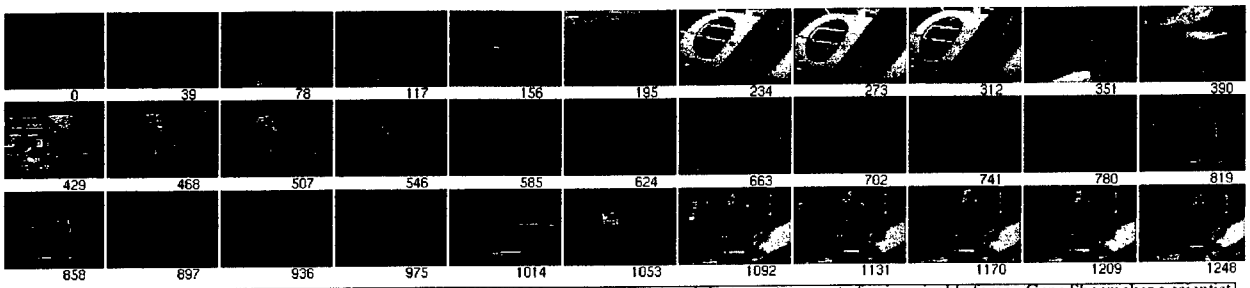
3.2 Video Skim Candidates

In order to create the image skim, we might think of selecting those video frames that correspond in time to the audio skim segments. As we often observe in television programs, however, the contents of the audio and video are not necessarily synchronized. Therefore, for each keyword or keyphrase we must analyze the characterization results of the surrounding video frames and select a set of frames which may not align with the audio in time, but which are most appropriate for skimming. To study the image selection process of skimming, we manually created skims for 5 hours of video with the help of producers and technicians in Carnegie Mellon's Drama Department. The study revealed that while perfect skimming requires semantic understanding of the entire video, certain parts of the image selection process can be automated with current image understanding. By studying these examples and video production standards [Smallman, 1970], we can identify an initial set of heuristic rules.

The first heuristics are the primitive rules, which are tested with the video frames in the scene containing the keyword/keyphrase, and the scenes that follow within at least a 5 second window. A description of each primitive rule is given in order of priority below. The four rows above "Skim Candidates", in Figure 7, indicate the candidate image sections selected by various primitive rules.

1. Introduction Scenes(INS)
The scenes prior to the introduction of a proper name usually describe a person's accomplishment and often precede scenes with large views of the person's face. If a keyphrase contains a proper name, and a large human face is detected within the surrounding scenes, then we set the face scene as the last frame of the skim candidate and use the previous frames for the beginning.

2. Similar Scenes(SIS)
The histogram technology in scene segmentation gives us a simple routine for detecting similarity between scenes. Scenes between successive shots of a human face usually imply illustration of the subject. For example, a video producer will often interleave shots of research between shots of a scientist. Images between



Keyphrases its restless geology erased the record - our long-dead Moon - by still more violent events - event of unimaginable force - Gene Shoemaker a scientist with the U.S. Geological Survey helped prove - another part of the evidence we can see - collisions with Earth are extremely rare - they reconstruct these great collisions - O.K., Earl, let's shoot it - and we can see the ejecta curtain simply move out - these clumps eventually will impact the surface - now this is the edge of the target - if we step back we find - this is what makes up the crater rays - and the Moon's dark plains - as astronauts roam the Moon's surface - most Moon rocks were between three and four billion - the samples gathered so carefully were not part - the early chapters of the Moon's history began to emerge

Figure 9: Skim video frames and keyphrases for "Planet Earth - I" (10:1 compaction).

window. The final skim borders are adjusted to avoid image regions that overlap or continue into adjacent scenes by less than 30 frames.

To avoid visual redundancy, we reduce the presence of human faces and default image regions in the skim. If the highest ranking skim candidate for a keyphrase is the default, we extend the search range to a 10 second window and look for other candidates. The human face rule is limited if the segment contains several interviews. Interview scenes can be extremely long, so we look for other candidates in a 15 second search window.

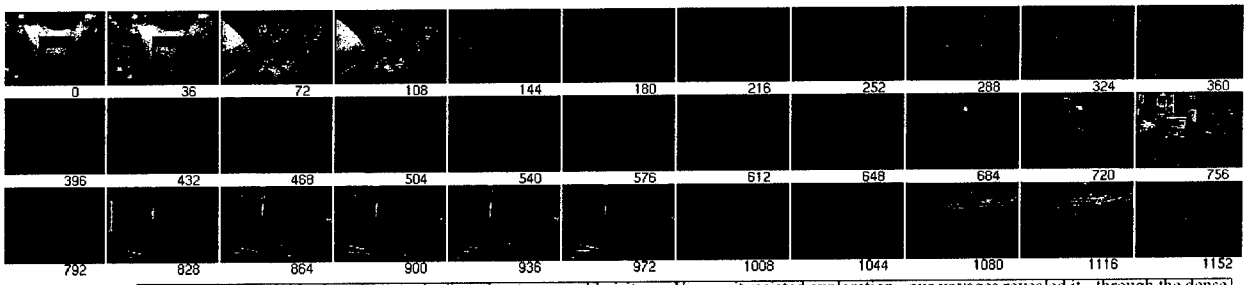
Figure 8 illustrates the adjustment and final selection of video skims. It shows how and why the image segments, which do not necessarily correspond in time to the audio segments, are selected.

3.4 Example Results

Figure 9 shows the video frames and audio from the "Planet Earth" video. The image portion of the skim has captured information from 18 of the 64 total scenes in the video. With the exception of the scene at frame 585, which lasts over 1,300 frames in the original video, most scenes are small and provide maximum visual information. An error in scene segmentation, near frame 702, causes this scene to split and, therefore, it is used twice for separate keyphrases. Introduction scenes, bounded camera motion and human faces dominate the image skims for this segment.

Figure 10 shows another example from the "Planet Earth" video with 16 of the 37 scenes represented. This segment contains many long outdoor scenes that provide little information. However, most primitive rules do not match these scenes so the search window is extended and they appear less frequently in the image skim. The scene at frame 828 is an interview scene which contains 3 keyphrases and lasts several frames. Even with an extended search window, the scenes that follow do not match any of the primitive rules so the image skim is rather long for this scene.

Figure 11 shows two types of skims for the "Mass Extinction" segment. Skim A was produced with our method of integrated image and language



Keyphrases to explore these neighboring planets - the first planet we would visit was Venus - it resisted exploration - our voyages revealed it - through the dense atmosphere winds up to two hundred miles an hour whip the clouds - no place can replicate the climate - but Death Valley, on a hot afternoon - how did it become the hot desolate planet - Jim Head, a planetary scientist at Brown University, is an authority - it's hot enough on the surface of Venus - the heat that comes into the surface can actually be - they radioed back images - and Jim Head and his colleagues had a global - we can also see some very distinctive highland - this one. Maxwell Montes - we have the same problem the resolution - Jim Head and his colleagues use a very special - the extensional forces are pulling

Figure 10: Skim video frames and keyphrases for "Planet Earth - II" (10:1 compaction).

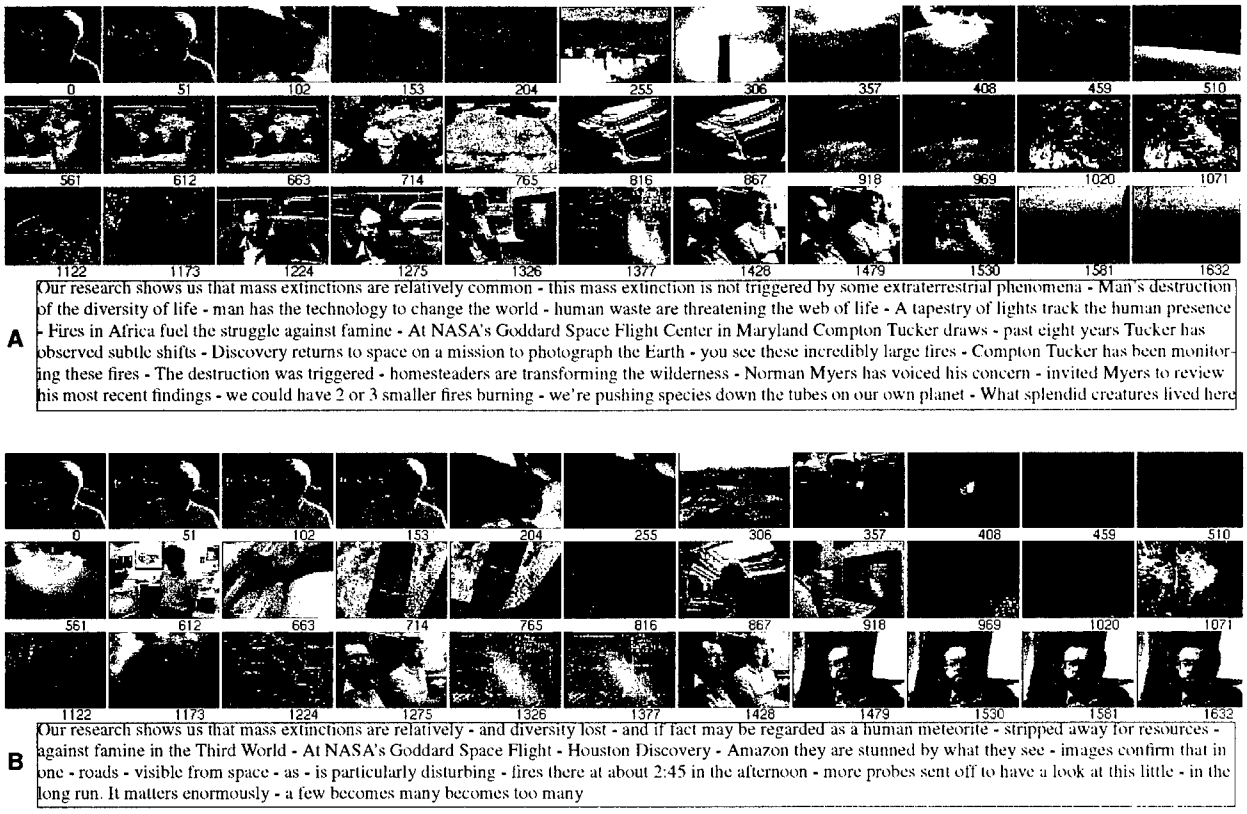


Figure 11: Image and text output for the "Mass Extinction" segment: A) Skim creation using image and language understanding, B) Skim creation using fixed intervals for image and audio.

understanding. Skim B was created by selecting video and audio portions at fixed intervals. This segment contains 71 scenes, of which, skim A has captured 23 scenes, and skim B has captured 17 scenes. Studies involving different skim creation methods are discussed in the next section.

Skim A has only 1632 frames, while the first scene of the original segment is an interview that lasts 1734 frames. The scenes that follow this interview contain camera motion, so we select them for the keyphrases towards the end of the scene. Charts and figures interleaved between successive human subjects are selected for the latter scenes.

3.5 User Evaluation

The results of several skims are summarized in Table 3. The manually created skims in the initial stages of the experiment help test the potential visual clarity and comprehension of skims. The compaction ratio for a typical segment is 10:1; and it was shown that skims with compaction as high as 20:1 still retain most of the content. Our results show the information representation potential of skims, but we must test our work with human subjects to study its effectiveness.

We are conducting a user-study to test the content summarization and effectiveness of the skim as a browsing tool in a video library. Subjects must navi-

Table 3: Skim Compaction Data

Title	Original(sec)	Skim (sec)	Comments
K'nex, CNN Headline News	61.0	7.13	MC-AS
Species Destruction I	68.65	6.40	MC-AS
Species Destruction II	123.23	12.43	MS
International Space University	166.20	28.13	MS
Rain Forest Destruction	107.13	5.36	MS
Mass Extinction	559.4	55.5	AC-AS
Human Archeology	391.2	40.8	AC-AS
Planet Earth I	464.5	44.1	AC-AS
Planet Earth II	393.0	40.0	AC-AS

Comments
MC - Manually Assisted Characterization
AC - Automated Characterization
MS - Manual Skim Creation
AS - Automated Skim Creation

gate a video library to answer a series of questions. The effectiveness of each skim is based on the time to complete this task and the number of correct items retrieved. Although our evaluation results are tentative, the skim does appear to be an effective tool for browsing, as evident by the difference of time that subjects spend in skim mode versus regular playback mode.

We use various types of skims to test the utility of image and language understanding in skim creation. The following creation schemes are presently being tested:

A - Image and Language Characterization
B - Fixed Intervals (Default)
C - Language Characterization Only
D - Image Characterization Only

Figure 11 shows examples of skim type A and B. The visual information in skim A is less redundant and provides a greater variety of scenes. The audio for skim B is incoherent and considerably smaller. Although our skim does appear to provide more information, additional testing is needed.

4 Conclusions

The emergence of high volume video libraries has shown a clear need for content-specific video-browsing technology. We have described an algorithm to create skim videos that consist of content rich audio and video information. Compaction of video as high as 20:1 has been achieved without apparent loss in content.

While the generation of content-based skims presented in this paper is very limited due to the fact that the true understanding of video frames is extremely difficult, it illustrates the potential power of integrated language, and image information for characterization in video retrieval and browsing applications.

Acknowledgments

We thank Henry Rowley and Shumeet Baluja for the face detection routine; Michael Witbrock and Yuichi Nakamura for the keyword selection routines. This work is partially funded by NSF, NASA, and ARPA. Michael Smith is supported by Bell Laboratories.

References

[Degen *et al.*, 1992] Degen, L., Mander, R., and Salomon, G. "Working with Audio: Integrating Personal Tape Recorders and Desktop Computers," *Proc. CHI '92,* 1992, Monterey, CA.

[Hampapur *et al.*, 1995] Hampapur, A., Jain, R., and Weymouth, T. "Production Model Based Digital Video Segmentation," *Multimedia Tools and Applications* **1** 1995.

[Mauldin, 1989] Mauldin, M. "Information Retrieval by Text Skimming," PhD Thesis, Carnegie Mellon University. 1989.

[Rowley *et al.*, 1996]Rowley, H., Baluja, S. and Kanade, K. "Neural Network-Based Face Detection," *Computer Vision and Pattern Recognition,* San Francisco, 1996.

[Wactlar *et al.*, 1996]Wactlar, H., Kanade, T., Smith, M., Stevens, S., "Intelligent Access to Digital Video: The Informedia Project," IEEE Computer, Vol. 29, No. 5, 1996

[Zhang *et al.*, 1993]Zhang, H., *et al*, "Automatic Partitioning of Full-Motion Video," *Multimedia Systems* 1993 1, pp. 10-28.

[Arman *et al.*, 1994a]Arman, F., Hsu, A., and Chiu, M-Y. "Image Processing on Encoded Video Sequences," *Multimedia Systems* 1994.

[Arman *et al.*, 1994b]Arman, F., *et al.*, "Content-Based Browsing of Video Sequences," *Proc. of ACM Multimedia '1994.*

[TREC, 1993] "TREC 93," *Proceedings of the 2nd Text Retrieval Conference,* D. Harmon, Ed., sponsored by ARPA/SISTO, 1993.

[Hauptmann and Smith, 1995]Hauptmann, A.G. and Smith, M, "Video Segmentation in the Informedia Project", IJCAI-95, *Workshop on Intelligent Multimedia Information Retrieval.* 1995.

[MPEG-1, 1991] "MPEG-1 Video Standard", *Communications of the ACM,* 1991.

[Smallman, 1970] Smallman, K., "Creative Film-Making", 1st ed., Publisher Macmillan, New York 1970.

Mixed Traffic and Automated Highways

Chuck Thorpe
Robotics Institute
Carnegie Mellon University

Abstract

A major issue in building a prototype Automated Highway System is whether the system needs dedicated lanes, occupied only by computer-controlled cars that communicate and cooperate with each other; or whether the automated vehicles can be provided with enough sensing and intelligence that they can safely operate in regular highways, intermixed with vehicles driven by people. A major portion of the CMU research effort in Automated Highways is focused on determining the technical feasibility of operation in mixed traffic. This paper outlines the issues of mixed traffic vs. dedicated lanes, then describes CMU efforts in building complete demonstration systems, vehicle sensors, obstacle sensors, car tracking software, reasoning for tactical driving, and deployment scenarios.

Mixed Traffic vs. Dedicated Lanes

The National Automated Highway Systems Consortium (NAHSC) is embarked on a seven-year project to build a prototype automated highway. The goal is to develop the specifications for a system that will allow completely hands-off and feet-off automated driving of specially-equipped cars, trucks, and busses, operating on specially-equipped lanes of high-speed limited-access roads. The AHS user will drive the vehicle normally on surface streets to the AHS entrance ramp, indicate a destination, then turn control over to the automated system, which will handle the driving until the right exit is reached.

We are in the middle of many important and interesting design studies: how should we handle obstacles? (detect them with onboard sensors? detect them with sensors built into the roadway? build strong fences and exclude all foreign objects?) Should automated vehicles platoon together, in tightly-linked groups of 10 vehicles, or should they only run as free agents, separated by 10 - 30 meters? What is the role of the driver: passive passenger, who will probably become complacent and distracted and therefore unavailable to help the automated driving system; or careful observer, able to spot subtle signs of potential obstacles?

Of all the design questions, perhaps the most interesting from a robotics viewpoint is whether the system should be based on dedicated lanes, or allow mixed traffic. The "dedicated lanes" approach means that vehicles will be allowed to operate under automated control only when in special lanes, physically separated from all manually-driven vehicles. The "mixed traffic" approach means that vehicles will be so capable of sensing and reacting to other vehicles, that they will be able to operate on freeways mixed in with human drivers.

The consortium as a whole is undertaking several studies to analyze the mixed and dedicated options separately, and then to compare the possibilities. At a high level, the discussion comes down to economics vs. technical feasibility. It is probably technically easier to build a dedicated lane facility. All the automated vehicles can be in communication with each other, running at the same speed, cooperating when a vehicle needs to change lanes, and sharing information about detected obstacles. But having a dedicated lane facility requires building one; and there is a chicken-and-egg problem of who will build the lanes before cars are available to use them; and who will buy the cars unless there are lanes on which they can run?

The mixed traffic option, on the other hand, would allow for relatively easy use of the entire network of freeways in the US. Some minor infrastructure may need to be added, depending on the technology used for lateral guidance, but at much lower financial cost than building new lanes, and probably at lower political cost than converting existing lanes for the sole use of

automated vehicles. Individuals who purchase a specially-equipped car could begin using it immediately, without having to wait for enough automated vehicles to be sold to justify having their own lane. The downside, of course, is the technical difficulty of driving in mixed traffic. The automated vehicles would have to be safeguarded against all the bizarre variations of human driving styles now encountered on the road.

Our group at CMU is most interested in investigating the feasibility of mixed traffic.[1] While the problems are difficult, the payoff for success would be large; and the kinds of questions that need to be addressed are important and interesting from a research standpoint. Even if the ultimate completely automated system does not become practical in the near term, the technology developed could play an important role in improving safety of partially-automated vehicles in the immediate future.

We are investigating mixed traffic feasibility on several fronts: building partially-capable demonstration systems; building vehicle sensors; developing car detection and tracking strategies; developing capabilities for tactical driving; and planning future development steps.

CMU Demo Vehicles

Some of the functionality of driving in mixed traffic has already been built for other purposes, and will be shown in August of 1997 at the NAHSC San Diego Demonstration. The 97 Demo is a congressionally-mandated "Proof of Technical Feasibility" for automated driving. Various members of the NAHSC will show a variety of capabilities, including both mixed traffic and dedicated lane driving as well as maintenance and inspection functions.

The part of the Demo to which CMU is contributing will emphasize independent sensing and decision making on board each vehicle, including the capability of driving in mixed traffic and also the ability to take advantage of communication with other intelligent vehicles in the vicinity. The demo scenario shows a mix of vehicles being driven manually, vehicles under full automated control, and partially-automated vehicles. The cars and buses will demonstrate lane departure warning and adaptive cruise control, as well as automated lane following, headway and speed maintenance, lane changing to pass slower vehicles, and obstacle detection and avoidance. When two automated vehicles are driving close to each other, they will communicate to share information about relative positions of themselves and of detected obstacles, so the trailing vehicle can safely drive with a smaller gap behind the lead vehicle. When automated vehicles are driving mixed with non-automated vehicles, they will automatically increase the free space buffer around themselves in order to have time to see and react to events.

The technology underlying the CMU portion of the demo starts with RALPH, the vision-based road following system built by Pomerleau.[2] RALPH resamples a video image to create an overhead projection of the road. In the overhead image, RALPH tests several hypothesized road curvatures to find the arc that most closely follows the dominant contrast features. This way, RALPH takes advantage of not only the painted stripes, but also the pavement joints, edge of the shoulder, and other features that run parallel to the road. Once RALPH finds the dominant curvature, it can look for lane boundaries and calculate the vehicle's lateral position in the lane. RALPH has accumulated over 25,000 km of road tests, including the "No Hands Across America" trip during the summer of 1995 during which it steered autonomously over 98% of the way from Washington DC to San Diego CA.

The demo vehicles are also equipped with forward-looking radar. The radars on the cars are provided by Delco Electronics. They are mechanically scanned in azimuth, to cover a 12 degree field of view. The radars provide range, bearing, and range rate to targets in front of the vehicles, and have integrated target tracking software to filter out spurious or inconsistent readings. Besides providing data to control separation from other vehicles, the radars are also capable of detecting obstacles that have enough radar reflectivity. The obstacles used for the 1997 Demo will be plastic construction barrels. In our initial tests, the radars can detect the barrels at up to 80 m, perhaps due to the reflective tape wrapped around the barrels.

The demo vehicles will also be equipped with side and rear looking sensors. The most difficult sensing requirement is forward, because stationary obstacles on the roadway need to be

detected at long ranges. Sideways sensing is relatively straightforward, and even rear-looking sensors for the demo scenarios need only have a range of a few tens of meters. Several sensors are currently being investigated for side and rear applications, including a variety of low-cost radars and sonars. The vehicles are also equipped with GPS positioning for navigation and for reporting the positions of detected obstacles.

The vehicles being built for the 97 Demo bring the Navlab family of vehicles up to 10. Navlab 1 is a Chevrolet van, now retired; Navlabs 2 and 4 are HMMWVs, mostly used for off-road driving research; Navlab 3 is a privately-owned Honda Accord, now returned to service as a non-automated car. The remaining vehicles are now or soon will be in on-road use. Navlab 6 and 7 are a matched pair of Pontiac Bonnevilles, designed for the 1997 Demo; Navlab 5 and 8 are minivans used for general experiments and driver warning studies; and Navlabs 9 and 10 are a pair of city busses, also being built for the 1997 Demo.

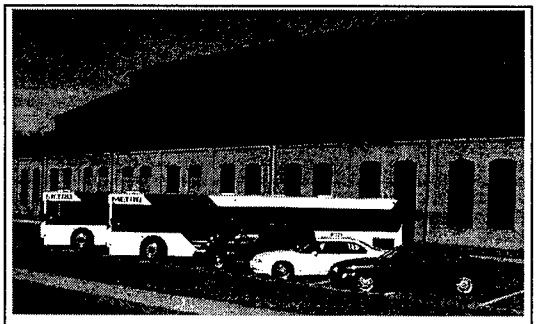


Figure 1: The Navlab Family, 6 - 10, front to back

Vehicle Sensing

The Demo system described above provides partial solutions for driving in mixed traffic, but are not yet adequate for full tests in unconstrained situations. The first requirement is for better sensing.

Dirk Langer's thesis work, completed in January of 1997, is one part of our effort.[3] Langer built a phased array radar that can cover a 12 degree field of view, with a range of 200 m, and does not use mechanical scanning. The specifications of the radar are:

- Range resolution: 0.6m
- Bearing resolution: 3 deg
- Range accuracy: 10 cm
- Bearing accuracy: 0.1 deg
- Repetition rate: 10 Hz

His software detects up to 20 radar targets in each measurement, and tracks those targets from measurement to measurement. The radar processing has been integrated with RALPH. The lane location and direction from RALPH are combined with the detected targets, to determine which targets are in the vehicle's lane and which are in adjacent lanes, even on curved roads. Similarly, the radar has also been integrated with GPS positioning and accurate maps to register targets with the next 100 meters of the road. This allows the radar to reject clutter such as guard rails or signs, while still properly detecting and reacting to stopped vehicles in the vehicle's own lane. The integrated systems have been demonstrated for a basic form of intelligent adaptive cruise control, and for detecting slow vehicles and triggering RALPH to change lanes.

Current work on the radar project includes redesigning the antennas for wider field of view and lower sidelobe intensity. The field of view in the current sensor was designed to accommodate normal freeway driving. A 16 degree field of view would be wide enough to handle standard exit ramps, and would allow detection of vehicles in adjacent lanes at closer ranges than the current system.

Obstacle Sensing

Beyond sensing vehicles, it is also important to sense obstacles on the roadways. This may be the most difficult technical challenge for automated driving; it is certainly the most difficult sensing challenge.

Obstacle detection is especially important for mixed traffic scenarios. Many of the obstacles found today on roadways come from other vehicles: the dominant source of debris is tire carcasses and retread, roughly followed by dead animals, spilled loads and dropped vehicle parts. (The dead animals were presumably alive when they wandered onto the roadway. In some parts of the rural US, the dominant cause of accidents is hitting deer). In dedicated lane configurations, some of these obstacles could be prevented by exercising more control over the vehicles on the roadway. It might be possible to inspect vehicles as they enter, and refuse entry to

vehicles that have unsecured loads; and it would be possible to continuously monitor the health of the vehicles, so if a tire began to overheat the vehicle could be slowed or stopped before the tire blew out. It may also be possible (although perhaps expensive) to build fences along the dedicated lanes to prevent animals from wandering onto the roadway. Also, when one vehicle detects an obstacle, it would be expected to notify other nearby vehicles of the location and classification of the obstacle. Finally, if there are relatively few miles of dedicated lanes, it might be possible to install sensors in the infrastructure, and communicate obstacle locations and suggested avoidance strategies to the automated vehicles. These strategies raise issues of feasibility, liability, and cost, but they are technically plausible and are all under study in the NAHSC.

For driving in mixed traffic, most of the obstacle exclusion or infrastructure sensing strategies are not feasible. The first automated vehicles on the road would encounter today's driving environment, with the same issues of dropped loads, shed retreads, stray deer, and so forth. Since other vehicles would not be automated, no particular help could be expected in finding and avoiding obstacles; although locations with particularly dangerous roadway configurations may need to be equipped with infrastructure-based sensors that can provide warning of obstacles out of sight around a corner.

Within the NAHSC, the first part of the work on obstacle detection is cataloguing the kinds of obstacles that are present. Some of this data is available from maintenance departments of state departments of transportation, and some is in the accident literature, but none of it has been carefully quantified. The second part if the problem is determining which of those objects are dangerous. Our colleagues at General Motors are conducting informal experiments to understand the effect on a vehicle caused by running over various objects. The vehicle may ride smoothly over the object, or the object may cause ride discomfort, or steering deflection, or structural damage. The next part will be to write careful specifications for obstacle detection sensing. Some parts of the specification are straightforward to calculate. The maximum range for obstacle detection is set by the stopping distance of typical vehicles. In the worst case the obstacle, roadway configuration, and adjacent traffic will conspire to prevent a lane change to avoid the obstacle, so the only possible maneuver will be to come to a complete halt. Other parts of the spec are much more troublesome. It would be convenient to define a radar cross-section for a typical obstacle, but while some objects have large radar cross-sections (dropped mufflers, steel-belted tire carcasses), others do not (wooden debris or deer. One of our colleagues hit a toilet that fell off a truck: porcelain has a very low radar cross-section).

At CMU we have started investigating possible obstacle detection methods even before the specifications are ready. One of the most promising approaches is using the reflectance channel of a ladar, being investigated by John Hancock as part of his thesis work. At the ranges of interest for obstacle detection (50 to 100 m), it is hard to generate a 3-D reconstruction of the roadway with enough accuracy to detect small objects (10 to 20 cm high). It may be more fruitful to look for changes in the reflectance of a patch of the road: even if the range is nearly the same as the ranges to the road plane, an object sticking up from the road will have a much lower viewing angle than the roadway, and will therefore reflect much more of the laser energy. Preliminary results are shown in Figure 2. A small object, in this case a chunk of wood approximately 10 cm high by 50 cm long, does not show up in the range data. In the reflectance channel, however, it is easily noticeable, and simple processing to extract different-looking patches from the road area easily finds the object.

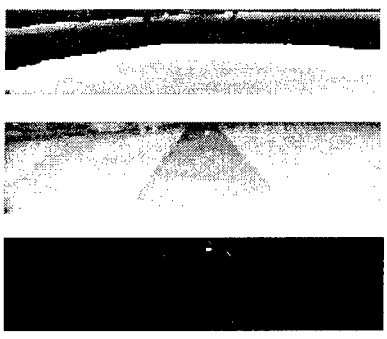


Figure 2: Obstacle Detection with Ladar Reflectance Top: range image. Middle: reflectance image, with obstacle near top of road. Bottom: Detected obstacle

We are investigating philosophically similar approaches for stereo processing. We have a real-time stereo machine, capable of generating 256 * 240 pixel depth maps at 30 Hz, using up to 6 input cameras. This means that standard stereo processing to find obstacles is possible in real time. But roadways are typically bland, without enough texture to generate high-confidence depth maps. Todd Williamson and John Hancock, as part of their thesis projects, are studying ways of detecting obstacles against bland surfaces. Part of the approach is based on confidence measures, such as those pioneered by Matthies.[4] If an image patch from the reference image matches all other images at some disparity with low error, then either the image patch is very bland or the patch is planar. If the image patch matches with high error, then the patch is probably both textured and non-planar. By making the windows to be matched large enough to cover both a road marking and a suspected obstacle location, it should be possible to detect objects by looking at the matching error. Again, as in the case of ladar reflectance processing, the presence of an obstacle would be sensed even without first doing a complete 3D reconstruction.

An additional observation is that stereo processing is normally set up to look for surfaces that are parallel to the image plane. If the cameras are all parallel and co-planar, then a rectangular window from one image matched against a rectangular window in another image at a given disparity implicitly defines a surface parallel to the images. We use an alternative approach, based on the projective stereo geometry popularized by Faugeras.[5] The CMU Stereo Machine has a lookup table for each pixel for each disparity. Using projective stereo calibration, the lookup tables can be set up to interpolate between any two given planes. By calibrating the stereo system with a ground plane and a higher plane, parallel to the ground plane (in practice, the surface of a campus loading dock), the disparity of each pixel in a source window is automatically indexed to match horizontal surfaces in the target images. This effectively skews the matching window so that a horizontal surface in the source image will be correctly registered with a horizontal surface in the target images. This should provide better results, since most of the world in front of the vehicle is nearly horizontal.

Car Tracking

Besides detecting obstacles, the ladar and stereo vision sensors can also be used for fine-resolution car tracking. Radar is good for detecting vehicles and reporting their velocity, but does not have fine enough resolution to generate an vehicle image. With ladar, the pixels are small enough, and closely enough spaced, that it is possible both to localize a vehicle within a lane, and to measure the orientation of the vehicle. Since cars steer non-holonomically, the vehicle orientation is an important cue of imminent lane changes.

The sensor we are using for these experiments is a scanning laser rangefinder built jointly by CMU and K2T Inc. The laser points up through the middle of the scan mechanism. The mirror is spun horizontally, and nodded vertically, providing 360 degree horizontal coverage and up to 35 degrees vertical field of view. Various laser rangefinders have been installed in the device, including a Riegel sensor with a 120 m range and 5 cm resolution. A new range sensor, built by Zoeller und Froehlich GmbH, will be installed shortly, and will have a pixel rate of up to 500 kHz.

The images in Figure 3 show range data from a car parked inside a building, processed by Liang Zhao. The data is first thresholded by elevation, to give just the data between 50 and 150 cm from the ground. The region where the car is expected is then processed to find straight lines, and finally the lines are fit to a model of the expected car shape. We are currently testing how much data needs to be collected on a car in order to do accurate localization. We will then build Kalman filters to integrate data taken from several scans as the vehicles move.

Tactical Driving

Most of the discussion to this point has been about sensing: how to see the road, see vehicles, detect obstacles, and track the course of other cars. Once the environment of the vehicle has been sensed, there still remain difficult and interesting problems in planning and acting.

Much of the automated vehicle literature has focused on the low-level problems of smooth control. Another set of research has worked on problems of route planning and guidance. There remains a hole in between these levels, which we

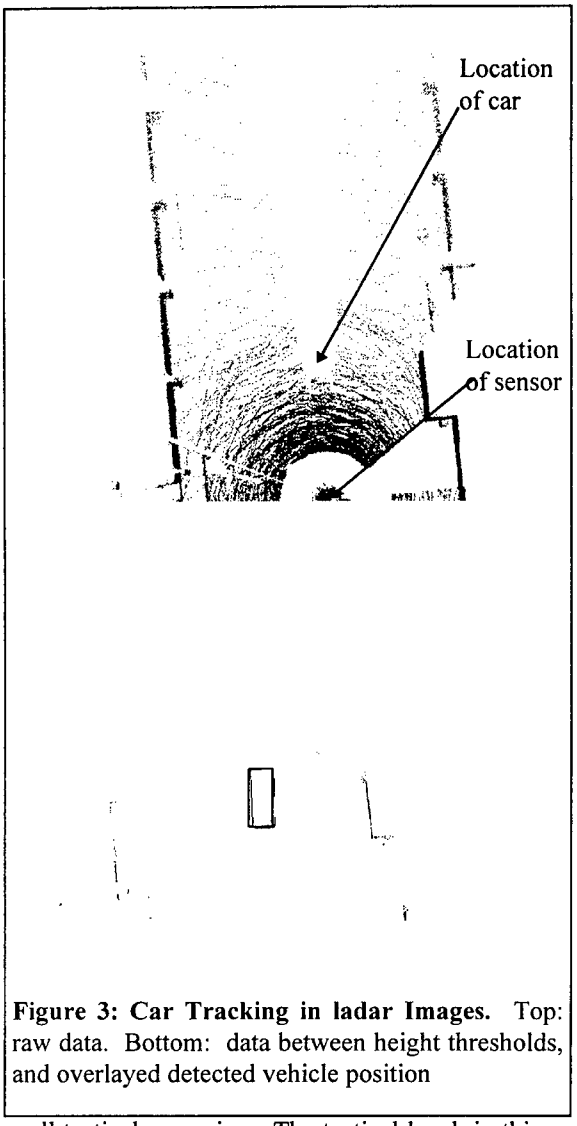


Figure 3: Car Tracking in ladar Images. Top: raw data. Bottom: data between height thresholds, and overlayed detected vehicle position

call tactical reasoning. The tactical level, in this case, refers to decisions about when to change lanes, when to speed up or slow down, how to trade off caution with making adequate progress, and so forth.

Rahul Sukthankar's recently completed thesis is a first step towards building safe and competent tactical driving.[67] For obvious reasons, his research was conducted in the framework of a simulator. His system, called SAPIENT, evolved through two main stages.

MonoSAPIENT is a single-threaded rule-based driving system. Sukthankar began his thesis work by reading the literature on driver's education and on defensive driving, looking for rules of safe driving. Unfortunately, this literature is not usually written in a form suitable for direct computer implementation. It is full of advice such as "when passing, select a lane relative to your speed, maneuvers, and traffic flow" which is undoubtedly sound, but impossible to directly operationalize. Instead, MonoSAPIENT's rules needed to be created from first principles (keeping a desired headway based on reaction times and braking performance), from rules of the road, and from experience. The resulting collection is turned into a decision tree, which can be interpreted in real time, to show good driving behavior for several vehicles in real time on a graphical simulator.

The difficulty with MonoSAPIENT is that the rules all interact with one another. This makes adding a new rule difficult. When a rule such as "pass the vehicle in front of you if it is going more than 5 mph under your desired speed" is added to the system, it competes with rules that say "prefer to stay in your own lane" and "do not pass if near the desired exit". In MonoSAPIENT execution, the relative priorities of the rules is determined by their place in the decision tree. Manually constructing those trees and ensuring correct ordering is difficult. Moreover, building specific rules for each specific situation is infeasible (e.g. "if the vehicle in front is going X slower than your desired speed and you are Y away from the exit and the traffic density is Z then it is OK to pass").

Partial solutions to these problems are developed in PolySAPIENT. Instead of a single set of rules, PolySAPIENT provides a separate "reasoning object" for each physical or logical object in the environment. Thus, each nearby car will have a reasoning object that keeps track of that vehicle, and the separation and relative velocity between that vehicle and the automated vehicle. Separate reasoning objects track lanes, exits, and internal parameters such as desired speed. Each reasoning object, at each time step, generates votes for desired actions and against bad actions, where the actions include both speed and turn commands. A knowledge-free arbiter selects the best action by a weighted combination of all votes.

PolySAPIENT is much more flexible than MonoSAPIENT. A PolySAPIENT vehicle can make intelligent trade-offs. While in MonoSAPIENT the rules are binary ("do not pass if ..."), in MonoSAPIENT the individual reasoning objects can cast graded votes for and

against actions. The result is that if several reasoning objects vote strongly for an action, and one or two reasoning objects vote weakly against it, the vehicle can choose that action. Thus PolySAPIENT vehicles are willing to squeeze into slightly tighter spaces than MonoSAPIENT vehicles, with a small sacrifice in desired headway, in order to move to a faster travel lane or to make a required exit. PolySAPIENT is also easier to program. While in MonoSAPIENT, all rules were implemented in the same style, in PolySAPIENT, the internal reasoning process of each reasoning object can be implemented with whatever internal structure is most appropriate. Also, new PolySAPIENT reasoning objects can be added at will, since each reasoning object is independent.

The disadvantage of PolySAPIENT is that tuning all the relative weights of votes from all the reasoning objects and setting internal parameters is a difficult process. Fortunately, the tuning process can be automated. Sukthankar expressed the weights and parameters to be tuned as a string of bits, then used PBIL, an evolutionary algorithm, to tune the weights and parameters.[8] Simulated vehicles are generated with their weights and parameters set probabilistically according to the current bit string. The vehicles are run through a series of simulated scenarios, and are rated according to criteria such as avoiding near-misses, arriving at their desired exits, and making adequate progress. The bit string is updated to more closely resemble the highly-ranked vehicles, and the process repeats. After approximately 20 generations, the vehicles learn to drive smoothly and safely.

Next Steps

The individual components of our research are all coming together. The vehicles and the core road following will be demonstrated in August 97; radars are becoming available and functional; obstacle detection is progressing; and the rules for tactical driving are running well in simulation.

The next big step is integration. We have already put two radars on at least one vehicle, so we can look forwards and backwards. We can probably treat obstacle detection separately, and not integrate that for the time being. We will put the new ladar scanner on the test vehicle, with the car detection and tracking software. We will enhance our car tracking Kalman filter to input data from both the ladar and the radar. Then we will test the complete system by driving manually and comparing our observations of vehicle positions around us with the sensed and filtered estimates.

Once we are happy with sensing, we can begin testing the SAPIENT driving strategies. At least at first, we will have SAPIENT generate recommendations, and watch to see if we drive the way it would drive. Later, we can have SAPIENT generate recommendations via a head-up display or speech synthesizer, so we can determine if the recommendations are safe and reasonable. If SAPIENT's advice does not follow our driving patterns, then a variant of the learning methods used in PolySAPIENT could be used to tune the weights to better match our own preferred driving styles. Once we are happy with the way the system works, we might enable SAPIENT control in stages, first giving it longitudinal control, then lateral control within a lane, then lane-changing abilities. Throughout, we have designed our systems to have easily-accessible kill switches and low-powered actuators so the safety driver can always override the automated control.

Acknowledgments

Our partners in the NAHSC Consortium are: Bechtel, Caltrans (the State of California department of transportation), Delco, the University of California PATH program, Hughes, General Motors, Lockheed Martin, Parsons Brinkerhoff, and the US Department of Transportation.

The Concepts work, within the Consortium, is the part of the project studying design tradeoffs, including mixed traffic vs. dedicated lanes. The first phase of the Concept work was led by Jim Lewis of Hughes, the second phase by Steve Schladover of PATH, and the third phase is being run by Steve Carlton of Lockheed Martin.

The CMU AHS group is Parag Batavia (vehicle sensing), Michelle Bayouth (Concept studies lead), Frank Dellaert (tactical driving), Dave Duggins (project manager), John Hancock (obstacle sensing), Martial Hebert (obstacle sensing), Todd Jochem (Demo lead), Katsumi

Kimoto (motion sensing), Phil Koopman (reliability and concepts), John Kozar (vehicle construction), Bala Kumar (motion sensing), Dirk Langer (radar), Sue McNeil (Societal and Institutional Issues), Illah Nourbakhsh (cooperating vehicles), Dean Pomerleau (vision for driving), Rahul Sukthankar (tactical driving), Toshi Suzuki (obstacle sensing), Todd Williamson (obstacle sensing), and Liang Zhao (vehicle tracking).

Funding for this work comes from the US Department of Transportation, agreement DTFH61-94-X-00001, "Automated Highway System". The ladar hardware was funded by the Ben Franklin of Pennsylvania Technology Partnership through contract 95W.CC005OR-2, and the first radar prototype by DARPA through TACOM in contracts DAAE07-90-C-R059 and DAAE07-96-C-S075, "CMU Autonomous Ground Vehicle".

References

[1] Bayouth, M., and Thorpe, C., "An AHS Concept Based on an Autonomous Vehicle Architecture", in Proceedings of 3rd Annual World Congress on Intelligent Transportation Systems, 1996.

[2] Pomerleau, D. and Jochem, J. (1996) Rapidly Adapting Machine Vision for Automated Vehicle Steering. IEEE Expert, Vol. 11, No. 2

[3] Langer, Dirk, "An Integrated MMW Radar System for Outdoor Navigation", Carnegie Mellon University, Jan 1997

[4] Matthies, Larry. Stereo Vision for Planetary Rovers: Stochastic Modeling to Near Real-Time Implementation. International Journal of Computer Vision, 8:1, pp. 71-91, 1992.

[5] Faugeras, Olivier. Three-Dimensional Computer Vision: A Geometric Viewpoint. MIT Press, Cambridge, MA. 1993.

[6] Sukthankar, Rahul, ""Situational Awareness for Driving in Traffic", Carnegie Mellon University, January 1997.

[7] Sukthankar, R., Hancock, J. and Thorpe, C Tactical-level Simulation for Intelligent Transportation Systems.. In Mathematical and Computer Modelling, Special Issue on Intelligent Transportation Systems, 1997.

[8] Baluja, S., Sukthankar, R., and Hancock, J. Prototyping Intelligent Vehicle Modules Using Evolutionary Algorithms. In D. Dasgupta and Z. Michalewicz, editors, to appear in "Evolutionary Algorithms in Engineering Applications", Springer-Verlag, 1997.

Optic Flow Estimation from 3D Wavelet Edge Detection

Andrew Lundberg* **and Lawrence B. Wolff***
Computer Vision Laboratory, Department of Computer Science
The Johns Hopkins University, Baltimore MD 21218
{lundberg, wolff}@cs.jhu.edu

Abstract

In this preliminary work, we've applied an edge detection wavelet transform to the problem of detecting motion/texture surface planes in a 3D image sequence space for the purpose of making fast optic flow measurements. The wavelet we've used is Mallet's spline based edge detection wavelet method [10], extended to 3D. Storage space considerations for the 3D wavelets and response to fine details are balanced to allow the method to detect optic flow in highly textured regions, without needing unreasonable resources.

1. Introduction

In the analysis of image sequences or processing of real-time visual input, image motion is one of the primary analytical measurements. In particular, observer motion (egomotion) and object tracking both require accurate optic flow calculations. Real-time applications of optic flow also require optic flow calculations at camera frame rates.

There have been many approaches for calculating optic flow [2]. These range from pattern matching between successive images [1], to filtering images sequences by sets of tuned filters [6][7]. There has also been some work on computing optic flow using wavelet filter banks [3]. The accuracy of these methods has improved greatly, but speed is still an issue. Wavelet edge detection methods hold promise for providing very fast optic flow calculations while handling multiple speed optic flows gracefully.

2. Wavelets and Edge Detection

The wavelet transform is a generalization of the Fourier transform, in that it transforms data into a coordinate system where many types of analysis are simplified. While the Fourier transform uses sine and cosine functions as it's basis, the wavelet transform can use any basis functions. Most applications of the wavelet transform require that these basis functions be carefully selected to guarantee orthogonality, as this makes the wavelet transform invertible. For a detailed treatment of wavelet orthogonality constraints, see Daubechies [5]. However, for applications where the wavelet transform is not an intermediary data representation, orthogonality is not a necessity. Edge detection, and optical flow from 3D edges are such applications.

Wavelets, which use short basis functions, are also differentiated from Fourier methods in that they are much better suited to isolating high frequency data such as edges. Because the Fourier basis functions are infinite in length, the Fourier response to a time/space data discontinuity is spread across all frequencies. In the wavelet domain, response to a discontinuity is spread across the multiple response resolutions, but the time/space localities are preserved. In addition to Mallat [10], other researchers have also developed wavelet edge detectors [3][8]. The wavelet transform for edge detection is attractive, in that it offers a multi-resolution paradigm for doing multiple pass edge detection as suggested by Canny [4].

The spline based wavelet we used consist of 2 functions, a low-pass smoothing function with 4 coefficients [1/8,3/8,3/8,1/8], and a high pass filter with 2 coefficients [+2,-2].

Because these functions have very few coefficients, the edge detection wavelet transform can be computer very quickly. It's also obvious when looking at these functions, that the wavelet coefficients at any single resolution are very similar to a Sobel operator, having a differential operator in one dimension, and a smoothing function in the other dimension.

Application of the standard wavelet transform is a

* This research was supported in part by ARPA grant DAAH04-94-G-0278, AFOSR grant F49620-93-1-0484 and the NSF National Young Investigator Award IRI-9357757

simple matter of convolving a decimated signal with the low pass and high pass wavelet functions, and then recursively processing the smoothed portion.

2.1 3D Wavelet transforms

Wavelet transforms in multiple dimensions can performed as separable 1D operations. To do a 2D wavelet, we do 1D wavelet transforms on each rows of the image, and then similarly transform the image column-wise. When processing successive coarser resolutions, we limit the range of the rows and columns to the smoothed data only, leaving the high pass results from the previous iterations intact. The 2D wavelet transform thus has 3 quadrants of high pass wavelet responses, x, y, and xy at each resolution.

For the analogous 3D wavelet transform, we have 7 octants at each resolution, containing the x, y, t, xy, yt, xt, and xyt surface responses. For determining optic flow, we use only the x, y, and t (time) surface responses. As in the 2D figure above, the multi-directional outputs are not smoothed in N-1 dimensions and thus generally have poor response.

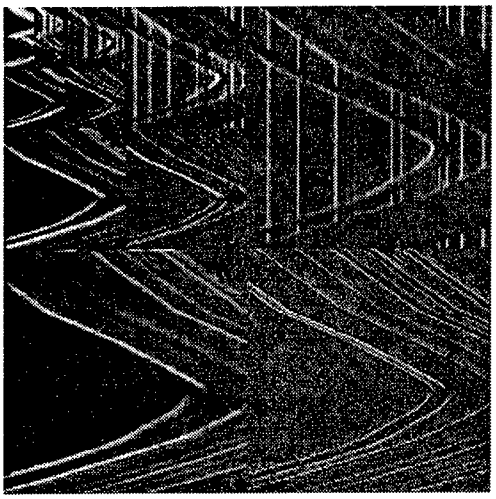


Figure 1: A 2D wavelet transform of a spatio-temporal slice from an image sequence. Note the separate horizontal/time edge responses in the upper right and lower left quadrants at each resolution.

3. Optic Flow from edge responses

An optic flow estimate is computed at each pixel on image from an image sequence. Optic flow is the 2D measure of how local image elements move between successive frames. This can be from objects moving through the image volume, or due to changes in image viewpoint.

At each single resolution, we estimate the optic flow from the x, y, and t edge responses. The x,y,t triple form a normal vector to the 3D edge gradient. For areas without optic flow, the edge gradient normal vector will be perpendicular to the time (T) axis, i.e. the t response will be zero. For high speed optic flow, the edge gradient will have a high ratio of t/x and t/y. Given a x,y,t response triple, we calculate the x and y portions of the optic flow vectors based on the following geometry:

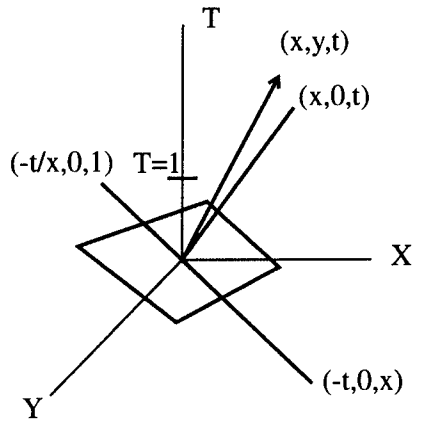


Figure 2: An image surface plane with resulting (x,y,t) response vector.

The optic flow is the projection of the 3D image surface plane onto the x,y plane at T=1. To find the X component of the optic flow, we first find the intersection of the X,T plane with the surface plane. (x,0,t) is the simple projection of (x,y,t) onto the x,z plane.

$$(x,0,t) \times (0,y,0) = (-ty,0,xy) = y(-t,0,x)$$

Solving for T=1:

$$(-t/x,0,x/x) = (-t/x,0,1)$$

So our X component of optic flow is simply -t/x and the Y component is similarly -t/y.

3.1 Multi-dimensional synthesis

The 3D wavelet transform creates a 3D sample space of responses to the wavelet filter. Three octants (at each resolution) of this space contain the x, y, and t surface orientation responses. These octants have been high pass filtered in one direction, and smoothed by the low pass filter in the other two dimensions.

The 3D wavelet transform is performed on a sequence of images containing a focal image for which we calculate optic flow. The time depth for this series is determined by the desired depth of wavelet transform recursion. For a depth K transform, we need at least a 2^K stack of images.

The x, y, and t single orientation octants (again, for each resolution) contain planes of edge responses for the focal image. These are combined as described above to determine x,y optic flow estimates at each resolution.

To combine these multi-resolution responses into a single optic flow map, we select from the optic flow estimates by selecting for a maximum confidence measure. As these estimates are derived from edge responses, we use the edge strengths as confidence measures. However, not all multi-resolution responses are equivalent. The effect of multiple smoothing operations on the edge data scales the resulting edge strengths [10]. These can be corrected, as they are constant effects at each resolution based on the scaling function of the wavelet. Mallat provides a set of these correction factors for his wavelets, and these can also be determined experimentally for any desired wavelet by comparing the multi-resolution edge responses to a simple step edge image.

4. Experimental Results

We collected real data of a patterned cup being moved across the image scene. The 3D stack of images were then transformed into the wavelet space, providing 3D edges at each image location. The 3D edge vectors were then transformed into 2D optic flow vectors.

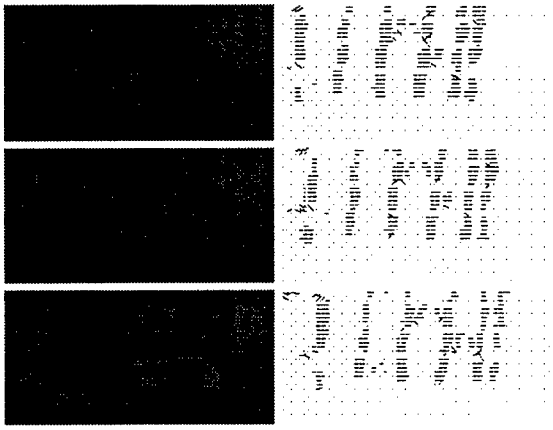


Figure 3: Frames from an image sequence of the chess cup and resulting optic flow vectors calculated from the 3D edge surfaces.

The optic flow calculations here while fast, are still quite coarse. As is clear from the outputs on the curved portion of the cup handle, these simple optic flow estimations from 3D gradient are subject to the aperture problem. That is to say, when looking at a local neighborhood (a small aperture view) we can detect motion across and edge, but cannot accurately determine motion speed components along that edge.

5. Conclusions

It's clear that multi-resolution gradient information be used to can detect regions of optic flow. Using multiple resolutions allows us to measure and integrate multiple speeds of optic flow. The method, as presented, does still have the limitation of only detecting optic flow components normal to image edges. Further work is necessary to find methods determine an accurate optic flow map from these edge gradient components.

References

1. Anandan, P. "A Computational Framework and an Algorithm for the Measurement of Visual Motion," *International Journal of Computer Vision* 2, pp 283-310, 1989.

2. Barron, J., Fleet, D., and Beauchemin, S. "Performance of Optic Flow Techniques," *International Journal of Computer Vision*, Vol 12, No 1, pp 43-77, 1994.

3. Burns, T., Rogers, S., Oxley, M., Ruck, D. "Computing Optical Flow Using Discrete, Spatio-Temporal, Wavelet Multiresolution Analysis, *SPIE Wavelet Applications*, Vol 2242, pp 549-560, 1994.

4. Canny, J. "A Computational Approach To Edge Detection," *IEEE Trans. Patt. Anal. Machine Intell.*, Vol 8 pp 679-698, 1986

5. Daubechies, I. "Ten Lectures On Wavelets," *CBMS-NSF Series Applied Mathematics*, SIAM 1991.

6. Fleet, D., and Jepson, A. "Computation of Component Image Velocity from Local Phase Information," *International Journal of Computer Vision*, Vol 5, No 1, pp 77-104, 1990.

7. Heeger, D. "Model for the Extraction of Image Flow," *Journal of the Optical Society of America*, A 4, pp 1455-1471, 1987.

8. Hevenor, R., Margerum, E. "Edge Detection Using a Complex Wavelet," *SPIE Wavelet Applications*, Vol 2242, pp 888-896, 1994.

9. LiuHui, LongGong, and TanZheng. "Edge Detection Using Adaptive Scales Wavelet Transform," *SPIE Wavelet Applications*, Vol 2242, pp 897-902, 1994.

10. Mallat, S. and Zhong, S. "Characterization of Signals from Multiscale Edges," *IEEE Trans. Patt. Anal. Machine Intell.* Vol 14, No 7, pp 710-732, July 1992.

SECTION II
IMAGE EXPLOITATION
(IMEX)

IMAGE EXPLOITATION
(IMEX)
PRINCIPAL INVESTIGATOR REPORTS

The RADIUS Phase II Program

Anthony Hoogs, Bill Bremner and Doug Hackett
Lockheed Martin Management and Data Systems
P.O. Box 8048
Philadelphia, PA 19101
[hoogs|bremner|hackett]@mds.lmco.com

Abstract

The Research and Development for Image Understanding Systems (RADIUS) Phase II program is described from the prime contractor's perspective. Intended to improve Imagery Analyst (IA) productivity through image understanding (IU) technology development and integration, the RADIUS program is centered upon the RADIUS Testbed System (RTS). The RTS enables the analyst to perform 3-D model-supported exploitation with IU assistance, and to construct 3-D site models using IU tools. Developed over the past three years, the RTS has led to a number of insights and new developments in IU technology and software, such as a framework for enabling the IA to easily interact with IU systems. These accomplishments are discussed, as well as the capabilities of the system and issues encountered in creating a large software system that integrates research results from a number of diverse institutions. The RADIUS program has made substantial progress toward its goals, but much more remains to be completed; potential directions for future work are described.

1 Introduction

The RADIUS program has the fundamental goals of increasing IA productivity and improving the quality and timeliness of IA products. Model-supported exploitation (MSE) was the underlying concept developed, demonstrated, and evaluated in RADIUS. It includes the generation and use of two- and three-dimensional features extracted from overhead imagery, feature and site attributes, source data information, and associated processes, to generate displays and perform automated analysis functions.

The objectives for the second phase included showing the utility of MSE, the use of IU Technology in support of MSE, evaluating the RTS, stimulating IU technology community interest in imagery intelligence (IMINT) problems, providing RADIUS results to the IMINT community and system developers, and encouraging the use of MSE and IU technology in operational systems [Gerson and Wood, 1994].

To achieve these goals, the focus of RADIUS Phase II was the development of RTS. The RTS is a prototype 3-D exploitation workstation that incorporates MSE as the central information organizing principle. The RTS improves the timeliness and quality of image exploitation by making many forms of softcopy information readily available to the analyst at the workstation, including images, 3-D models of site features, text and attributes linked to site features, and numerous querying tools. In addition, IU algorithms are available to assist the IA in constructing site models, and performing *change detection* and *detection and counting* analysis tasks.

The RTS was designed to illustrate the utility of IU and MSE technologies in a near-operational environment. Phase I of RADIUS validated the use of MSE and IU technology by exposing IAs to concept demonstrations. Phase II implemented the concepts in a hands-on workstation system that enables analysts to experiment with the technology. The RTS has been used to test many aspects of image exploitation, including:

- end-to-end image exploitation
- batch processing of automated exploitation algorithms
- manual, semi-automatic and automatic site model feature extraction
- characterization of IU algorithms
- constructing complete softcopy site folders
- automated image-to-site-model registration
- multi-image registration
- Human-computer interface (HCI) design for IU-assisted exploitation
- database storage and representation of 3-D features
- data export/import of site model features
- image and data interfaces to image and textual intelligence databases
- integrating Lisp and C/C++ development

This spectrum of capabilities ranges from support for development of specific IU capabilities to IA-centric exploitation functions. It reflects the balance between testbed and IU technology development; the RTS was designed to facilitate IU integration while supporting an HCI that would enable IAs to evaluate the utility of MSE and IU within MSE.

Using the suite of site model construction and exploitation algorithms integrated into the RTS, formal IA evaluations of the system have begun. Preliminary indications are that IAs are willing to manually construct or edit 3-D models of site features, and that semi- and fully-automated site model construction (SMC) algorithms have promise, but need more research to achieve their potential.

RADIUS explored how IAs and IU algorithms interact to perform exploitation. As implemented in the RTS, the resulting *First-Look Paradigm* [Bailey et al., 1994] (also referred to as the *Quick-Look Paradigm*) enables an IA to specify automated tasks to be applied to site features in subsequent imagery [Bremner et al., 1996]. In this paradigm, algorithms are required to derive any image-specific parameter values, so that all processing happens in batch without further IA assistance [Mundy and Vrobel, 1994]. Exploitation results, presented in a unified, graphical user interface, (GUI) insulate the IA from the processing of IU algorithms.

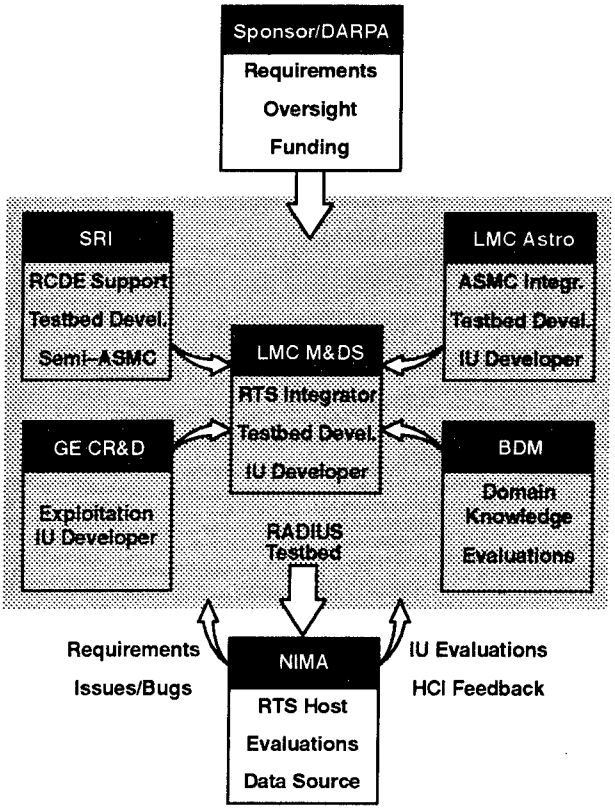


Figure 1: The organizations responsible for the RTS.

The selection of image data of intelligence events depicted over a range of operational conditions is critical to the evaluation and feedback of IU algorithms. While sufficient image data sets were provided to support experiments in site model construction, the resources available to support formal testing of change detection algorithms were insufficient to fully characterize algorithm performance.

The complete RADIUS Project has involved a variety of government, industrial and academic institutions. The RADIUS Phase II contract was primed by Lockheed Martin Management and Data Systems (Valley Forge, PA), with subcontracts to SRI International (Menlo Park, CA), General Electric Corporate Research and Development (Schenectady, NY), Lockheed Martin Astronautics (Denver, CO) and BDM Federal (MacLean, VA). The general roles of the team members and the primary government agencies are outlined in Figure 1. Other institutions not shown in the figure provided consultation, guidance and algorithms. Algorithm integration was performed by each contract team member, except BDM.

The RTS is installed at the former National Exploitation Lab (NEL), which is now part of the National Imagery and Mapping Agency (NIMA). The program sponsor, the Defense Advanced Research Projects Agency (DARPA) and NIMA have worked closely with the LMC contract team to develop and evolve the RADIUS concepts during Phase II as the RTS software development and integration was underway.

The next section of this paper summarizes the current capabilities of the RTS, including integrated IU algorithms. Section 3 describes the evolution of the system, tracking how the technology progressed during the program. This is followed by discussions of the major accomplishments and issues of the program, and future directions for RADIUS technology.

2 RTS Capabilities

The RTS contains many MSE capabilities that demonstrate how manual and automated exploitation can be integrated with and enhanced by site models. For manual exploitation, the RTS includes image manipulation tools, mensuration tools, cables, baseline text, links between images and site features, exploitation tools, site model updating tools, report generation support, and data input/output in several forms. Semi-automated and automated IU support is

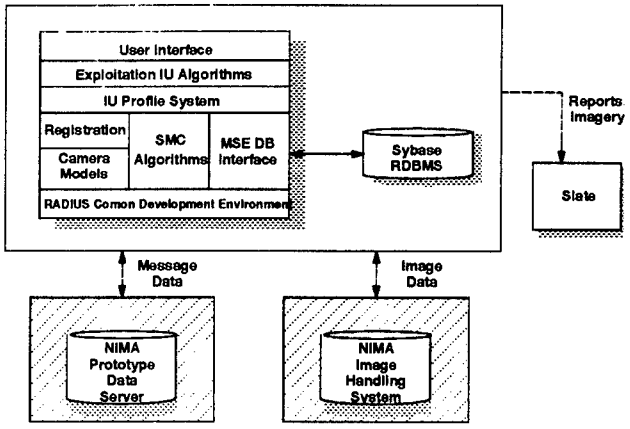


Figure 2: The major RTS components.

provided for detection and counting, change detection, and site model construction (feature extraction).

This functionality is built upon the RADIUS Common Development Environment (RCDE), as illustrated in Figure 2. The figure shows the major components of the system and links to external systems. All manual MSE capabilities are implemented through the RCDE and the MSE DB Interface [Hoogs and Kniffin, 1994], and accessed through the IA User Interface. Exploitation IU algorithms are integrated through the IU Profile System, or the Exploitation IU Framework [Bremner et al., 1996, Kniffin and Hoogs, 1996]. Site model construction algorithms operate through the RCDE, producing RCDE representations of extracted 3-D features.

The rest of this section briefly describes the functionality of the system, including basic capabilities, 3-D modeling, exploitation IU algorithms, and site model construction IU algorithms.

ELT Tools The RTS supports a variety of Electronic Light Table (ELT) functions. It displays 3-D site models overlaid on imagery, and enables the user to manipulate both for manual imagery exploitation. This includes standard image manipulation functions such as contrast stretch as well as roaming, zooming, and orienting the image/model display.

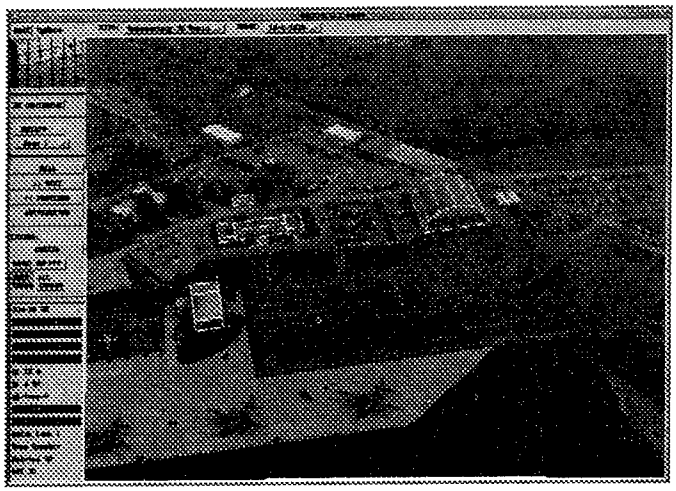


Figure 3: The Site Browser Panel.

The RTS registers 3-D site models to imagery to manipulate the imagery with reference to the model, such as zooming to fit the pixels associated with a 3-D region (e.g., a building) or automatically centering and zooming images to be viewed around a 3-D point. The RTS can rotate a displayed image so that vertical edges in the scene are vertical on the screen.

The 3-D wire-frame objects overlaid on the imagery are organized into separate graphic layers for convenience. Their appearance, such as color, texture, and line width, can be modified by the analyst. 2-D and 3-D annotations can be added to the scene to aid object identification and produce annotated images as output products. All ELT tools are accessed through the Site Browser panel illustrated in Figure 3.

Site Model Data In the MSE concept, the 3-D site model provides access to all the data related to a feature of interest. The RTS can import imagery, text, and other collateral data, storing them in a Sybase relational database for retrieval and feature-specific queries. Other data interfaces to and from the RTS include the NIMA Prototype Data System (NPDS) for retrieving site baseline text reports, cables, analysts' reports, and engineering support data; and the NIMA Image Handling System (IHS), for retrieving new imagery and creating National Imagery Transmission Format (NITF) 2.0 image pyramids.

Manual MSE One of the purposes of the RTS is to explore how interactive forms of MSE can yield improvements in analyst productivity without the use of IU technology. To investigate this, a number of manual MSE capabilities were implemented in the RTS.

Analysts can access site model data directly from the user interface by point and click on the feature of interest including collateral text, links and attributes. Collateral text couples text of a general nature to individual objects. Links allow specific associations between objects and images, and between objects and text descriptors. Attributes are object-dependent features that contain information such as the history, normalcy and material type.

The RTS also contains a set of tools to retrieve images from the database based on their collection parameters. The Viewpoint Query Tool allows the analyst to specify a desired viewpoint by rotating the site model to the desired perspective. The database is then queried for images with similar viewpoints based on refined azimuth and elevation values. In contrast, the Image Query Tool can be used to query the database for images by collection parameters such as geographic area, sensor, and date range. This is useful for performing the task of manual negation, in which the analyst can browse through a set of images in reverse chronological order and immediately discern when scene changes took place at some time in the past.

Having access to geolocated 3-D information greatly improves analyst tasks such as mensuration. 3-D Ruler objects have endpoints that can be placed accurately in multiple images or attached to site model objects (e.g., building corners) to obtain very accurate object measurements. Other standard tools are terrain contours, terrain grids, and user-controlled terrain visualization tools overlaid on the imagery.

Automated Image Exploitation A class of IU algorithms integrated into the RTS can be used for automatic imagery exploitation. RADIUS supported the integration of algorithms from a wide variety of university and industry

sources, which was made possible using a common testbed environment (the RCDE) and an *Exploitation IU Framework* that was developed under RADIUS.

Described in [Bremner et al., 1996, Kniffin and Hoogs, 1996], the IU Framework serves as an Application Programmer Interface (API) to the RTS for IU algorithms. It enables the IU developer to focus exclusively upon the development of special purpose IU algorithms that can be applied to a specific feature within a site. That feature can be small enough to allow simplifying approximations based upon local image context. For example, the pixel intensity distribution of a projected feature may be nearly uniform under normal circumstances. Deviations from uniformity may be considered to be a significant change.

The framework separates site-specific and image-specific information. Site-specific data, such as the feature of interest, is established once by the IA when an IU algorithm is chosen to operate on the feature, resulting in a *profile*. Algorithm-specific inputs are also gathered at this point. When new images are introduced to the RTS, they are processed by applying appropriate profiles without any IA input. Any image-specific information required by the algorithm must be automatically extracted.

The IU Framework also supports the specification of conditions under which a given special purpose algorithm will not perform well, including occlusions, shadows, and clouds. In addition, historical information about image conditions may be specified for dynamic reference and calculation of photometric corrections.

Finally, the framework provides for the storage of IU results in a standardized format in the RTS database. That format requires each algorithm to specify the degree of change or count, and the degree of confidence in that result, with a subimage showing a graphic overlay of the results on the feature.

The framework includes an extensive user interface, part of which is shown in Figure 4. The figure shows the history results panel and two images. In this case, a single profile was executed against sixteen images, generating sixteen IU results. In the larger panel, the results are listed in tabular form, displaying for each result (from left to right) the profile name, image identification, a yes/no indication of change, a numerical level of change between 0 and 1 (1 = complete change), confidence measure in the change level, and user-assigned profile priority. Below the table is a graph of the results, with images plotted in chronological order on the horizontal axis and the change level plotted on the vertical axis. The upper panel shows two of the sixteen images, with a building model overlaid. The profile reports no change if the building is present, and change if the building is not detected, which will occur if the building has been sufficiently renovated or damaged. The left image is correctly reported as no change (it is the seventh image in the list), and the right image is correctly reported as change (the fourth image in the list, corresponding to the tall peak in the graph). In the right image, change in the structure is simulated by misregistering the image to the site.

This comprehensive display provides the IA simultaneous access to textual, graphical, and site model views of IU results. This combination enables the IA to verify or refute IU computations, while examining a summary of multiple results over time, to establish trends or historical context.

Within the IU Exploitation Framework, all IU algorithms fall into two categories: detection and counting or change detection. The exploitation IU algorithms are generally accessed through the Monitoring, Negation, or Detection and Counting menus, for use in setting up profiles. Table 1 summarizes the task-based exploitation algorithms currently available in the RTS, and Table 2 lists supporting algorithms that are optionally applied before the exploitation algorithm in a given profile. Further information on these algorithms is available in [Mundy, 1996, Hoogs and Bajcsy, 1996, Hoogs and Bajcsy, 1995, Huttenlocher, 1993, Sarkar and Boyer. 1993, Chellappa et al., 1996,

Table 1: Exploitation IU Algorithms. Key for Task column: CD = change detection, DC = detection and counting. Key for Context column: H = historical reference used, R = registration refinement available, C = calibration through reference patches.

Algorithm	Source	Application	Task	Context
Edgel Change Detection	GE	Detects significant change in edge density.	CD	H, C, R
Man Made Structure Change Detection	GE	Detects new objects with predominantly linear structure.	CD	H, C, R
Line Orientation Change Detection	GE	Detects linear structures matching a specified orientation and position.	CD	H, C, R
Albedo-Based Change Detection	GE	Detects changes in the albedo (brightness or darkness) of a region.	CD	H, C, R
Perceptual Grouping Change Detection	OSU, USF, GE	Detects axially symmetric structures.	CD	C
Hausdorff Building Validation	Cornell, GE	Validates the presence of modeled features using the Hausdorff distance metric.	CD	R
Building/Structure Presence	LMC VF	Detects changes in or the removal of volumetric features using model geometry and prior imagery.	CD	H, R
Delineated Feature Presence	LMC VF	Detects changes in or the removal of planar features using model geometry and prior imagery.	CD	H, R
Building Validation	USC	Detects changes in or the removal of modeled buildings using model geometry and shadow information.	CD	R
Vehicle Detection and Counting	UMd	Detects vehicles in a delineated parking area using a generic rectangular model.	DC, CD	
Correlation Detection and Counting	LMC Denver	Matches instances of a user-selected example object in a region on the same EO image.	DC	
EO Correlation Counting	LMC Denver	Matches instances of an example object in a region by identifying a template derived from historical EO imagery.	CD, DC	R
SAR Region Match Detection and Counting	LMC Denver	Uses ARAGTAP algorithm to match instances of a user-selected, bright example object in a region on the same SAR image.	DC	
SAR Region Match Counting	LMC Denver	Uses ARAGTAP algorithm to match instances of a bright example object in a region by identifying a template derived from historical SAR imagery.	CD, DC	R
SAR Correlation Counting	LMC Denver	Matches instances of an example object in a region by identifying a feature template derived from historical SAR imagery.	CD, DC	R

Table 2: Exploitation Support Algorithms. Key for Context column: H = historical reference used.

Algorithm	Source	Application	Context
Model-Based Cloud Detection	LMC VF & Orlando	Detects cloud obscuration of individual features using model geometry and texture analysis.	H
Hierarchical Pose Refinement	LMC VF	Corrects for localized, translational image registration errors up to 50 pixels using modeled feature edges and prior imagery.	H
Hausdorff Pose Refinement	GE, Cornell	Corrects for localized, translational image registration errors up to 10 pixels using the Hausdorff distance metric.	

Table 3: Site Model Construction Algorithms. Key for Context column: S = semi-automatic, A = fully automatic, M = multiple images used.

Algorithm	Source	Application	Context
Snakes	SRI	Optimizes the geometry of a user-supplied initial model of a linear feature using multiple images.	S, M
Model-Based Optimization	SRI	Optimizes the geometry of a user-supplied initial model of a polyhedral feature using multiple images.	S, M
Zip-lock Snakes	ETH	Creates a model of a linear feature given two points at opposite ends of the feature.	S
Road Tracker	SRI	Extracts the geometry of a constant-width linear feature given two points on opposite sides of the feature.	S
Cookie Cutter	SRI	Detects and models identical buildings given a model and position of one building.	S
Hub	SRI	Determines the applicability of SMC algorithms given a set of images and other context.	
Automatic Site Model Construction	UMass	Extracts the position and geometry of all flat-roofed, rectilinear buildings in an area or site.	A, M
Automatic Site Model Construction	USC	Extracts the position and geometry of all flat-roofed, rectilinear buildings in an area or site.	A, M
Automatic Site Model Construction	CMU	Extracts the position and geometry of flat-roofed and peak-roofed rectilinear buildings in a site.	A, M
Automatic Road Extraction	Brown	Extracts the geometry of all linear features in an image by locating starting points and tracking from each.	A

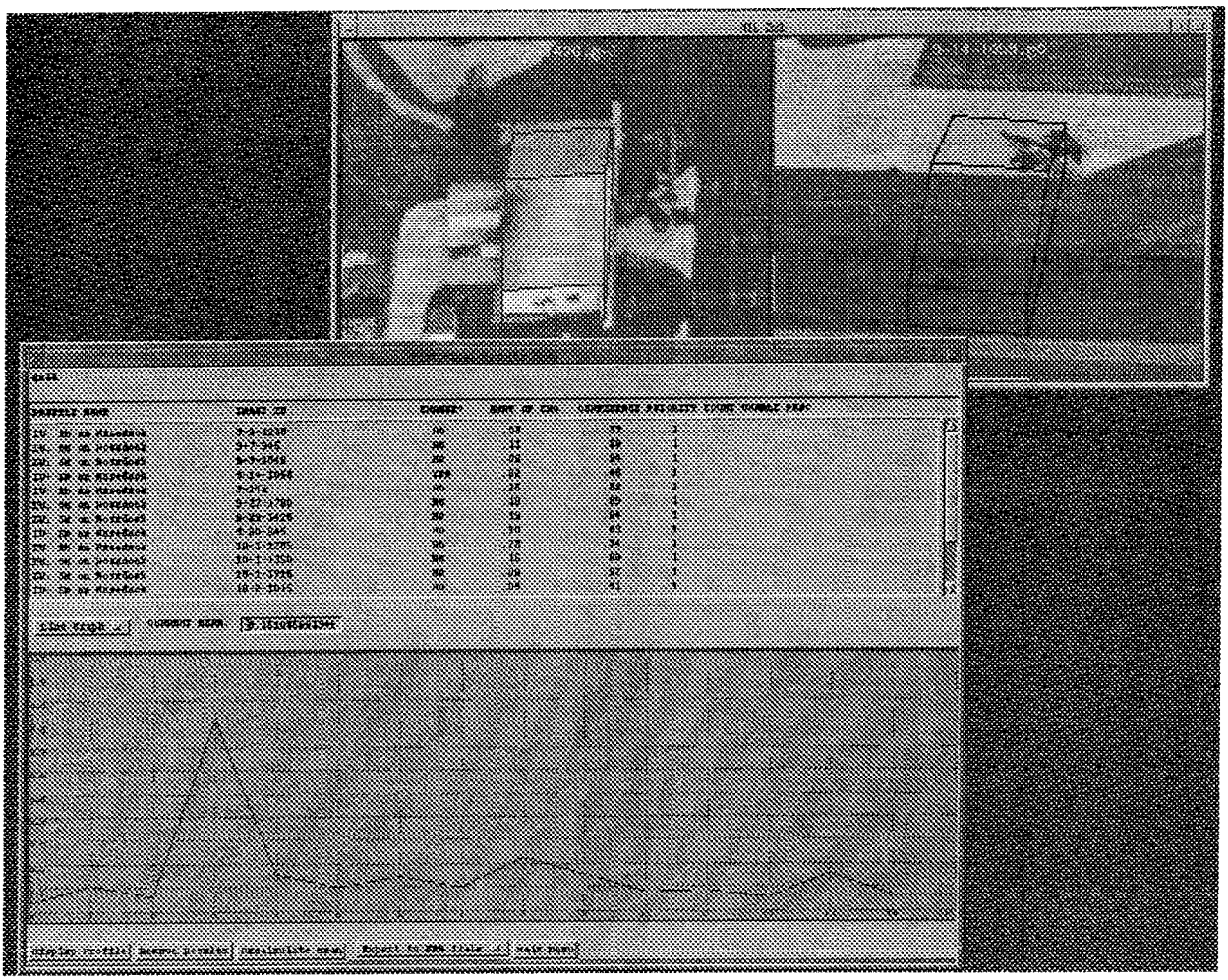


Figure 4: Trends and history display in the IU Framework.

Huertas and Nevatia, 1996].

Site Model Construction The RTS has major subsystems for the creation of 3-D site models containing complex volumetric and surface features. These tools enable manual, semiautomated, and automated techniques to configure a site model and populate it with features.

Construction of a site model is accomplished by 1) deriving a consistent, local geometry from imagery and 2) populating the space with models of site features. The local geometry is derived by collecting a set of images, placing a number of 3-D points in the scene, and simultaneously adjusting both the points and the image collection parameters to minimize the errors in projecting the 3-D points into all images. The site model is populated by placing models of features such as buildings and functional areas into the scene, attaching collateral text to the models, and adding model attributes.

The RTS has a variety of tools to support these activities. In particular, the Object Creation Tool, depicted in Figure 5, simplifies the process of adding new models to a site. Procedurally, the IA first selects a location for the object to be created. The image database is then queried for the four images with maximum mutual disparity, i.e., the images that show the location from widest selection of viewpoints. Next, the user picks an object type and models the object using manual and semiautomatic manipulation in the four-image display panel, which continuously updates all four images.

Unlike exploitation IU, there is no integrat-

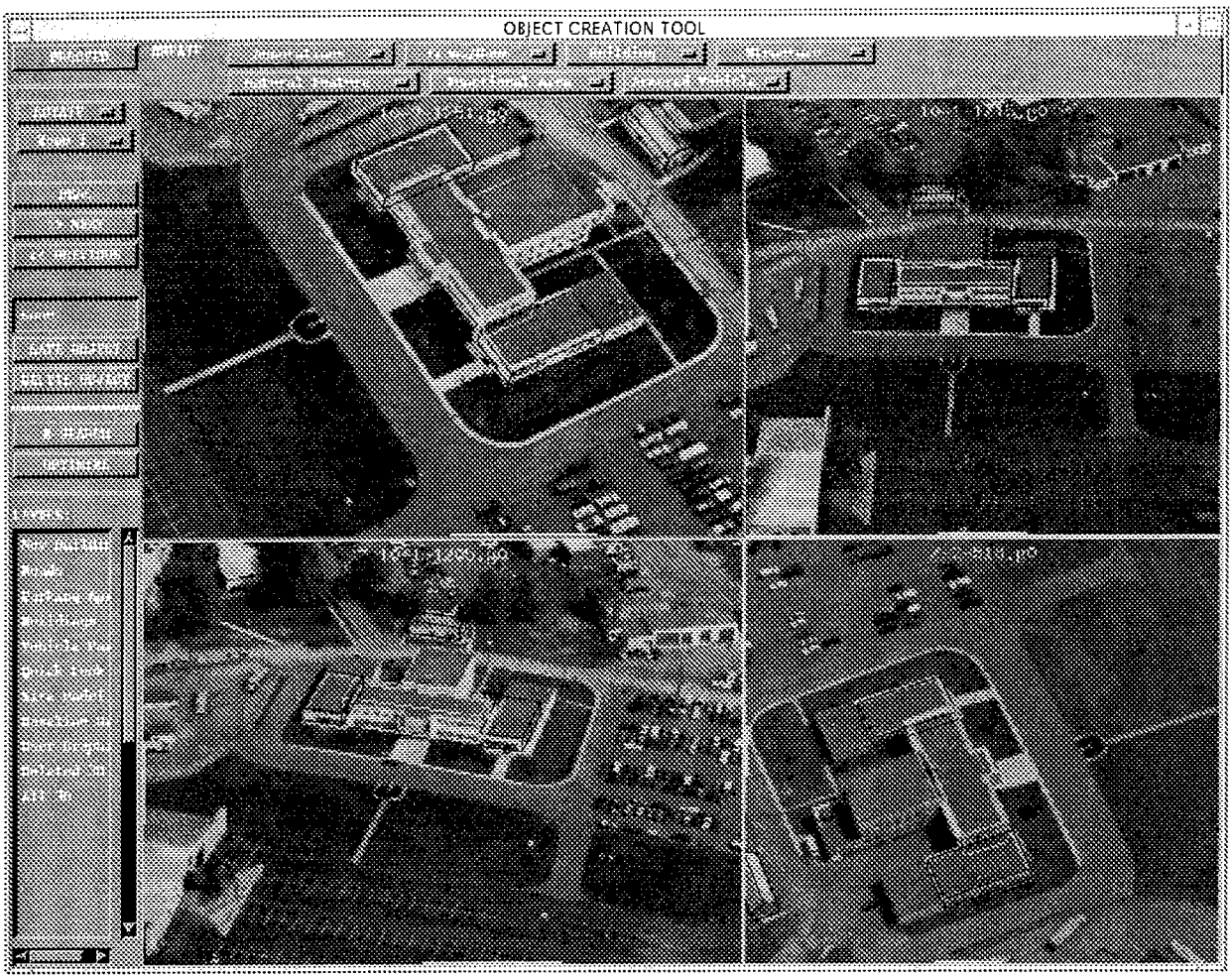


Figure 5: The Object Creation Tool.

ing framework for site model construction systems beyond the RCDE itself. SMC algorithms produce RCDE objects as output, and these models can then be displayed and edited by the IA. However, the Hub system can control the execution of SMC algorithms by enabling algorithm developers to encode algorithm constraints and characteristics in a rule-based framework. When the IA performs an SMC operation, the Hub suggests which algorithms are appropriate given the images being used and the algorithm rule base.

The suite of automatic and semiautomatic algorithms for site model population integrated into the RTS is listed in Table 3. The level of integration varies significantly, from real-time, interactive manipulation to file-based transfer between systems that must be operated independently by the user. These algorithms are described in detail in [Lin and Nevatia, 1996, Collins et al., 1995, Neuenschwander et al., 1994, Fua, 1996, Barzohar and Cooper, 1993, McKeown and Roux, 1994].

3 RTS Development and History

The design and implementation of the RTS is based on solutions to a number of significant challenges. The driving issues concerning the design included:

- the balance between a research and development environment, and a prototype exploitation workstation;
- ease of IU integration;

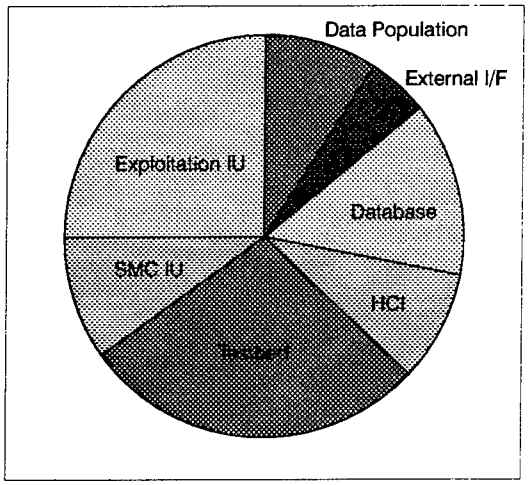


Figure 6: The proportion of technology development effort spent on major RADIUS tasks.

- ease of adding IA support, such as user interfaces and database queries.

While the IU components of the system are critical to its success, it is also important to provide a suitable user interface so that the IU capabilities can be perceived clearly. An awkward user interface can cause IAs to rate a system negatively, even when the IU has great potential for improving IA efficiency. On the other hand, much IU research and development was necessary to develop new algorithms that fully exploit the available context in the MSE framework.

The relative breakdown of technology development efforts by the RADIUS Phase II contract team is shown in Figure 6. Comprising the large majority of the total contract cost, these tasks were:

Exploitation IU: Evaluation and selection of algorithms from the IU community, and development and integration of the algorithms described in Tables 1 and 2.

SMC IU: Selection, development, integration, and evaluation of the semiautomatic and automatic site model construction algorithms listed in Table 3.

Testbed: Design, development, and integration of workstation capabilities not included in other categories.

HCI: Design and development of IA-oriented user interface capabilities, including consultation with NIMA and IAs.

Database: Design and development of the interface to and usage of Sybase, a commercial relational database management product. This work is described in [Hoogs and Kniffin, 1994, Kniffin and Hoogs, 1996].

External Interfaces: Design and development of interfaces to external systems, such as Slate (a reporting tool), the NIMA Image Handling System, and the NIMA Prototype Data Server.

Data Population: The creation and processing of site models and imagery.

The Data Population, HCI, and Testbed tasks developed into a larger portion of the effort than expected at the beginning of Phase II. Data Population in particular was hampered by the initial set of site models, which contained geometric inconsistencies. The development of sensor models and photogrammetric methods also consumed more resources than expected.

A significant amount of research in exploitation IU was required, since existing algorithms were not designed to take advantage of the level and types of context available in the MSE framework. This changed the expected ratio of algorithm development vs. integration to favor algorithm development.

The RTS has evolved as a software system over the past fifteen years, beginning with the SRI Cartographic Modeling Environment (CME), continuing with the RCDE, and culminating in the RTS. Figure 7 shows this development path over the past ten years (the previous five contained early CME development), and summarizes the capabilities that each stage added to the system. Parts of other major systems are in-

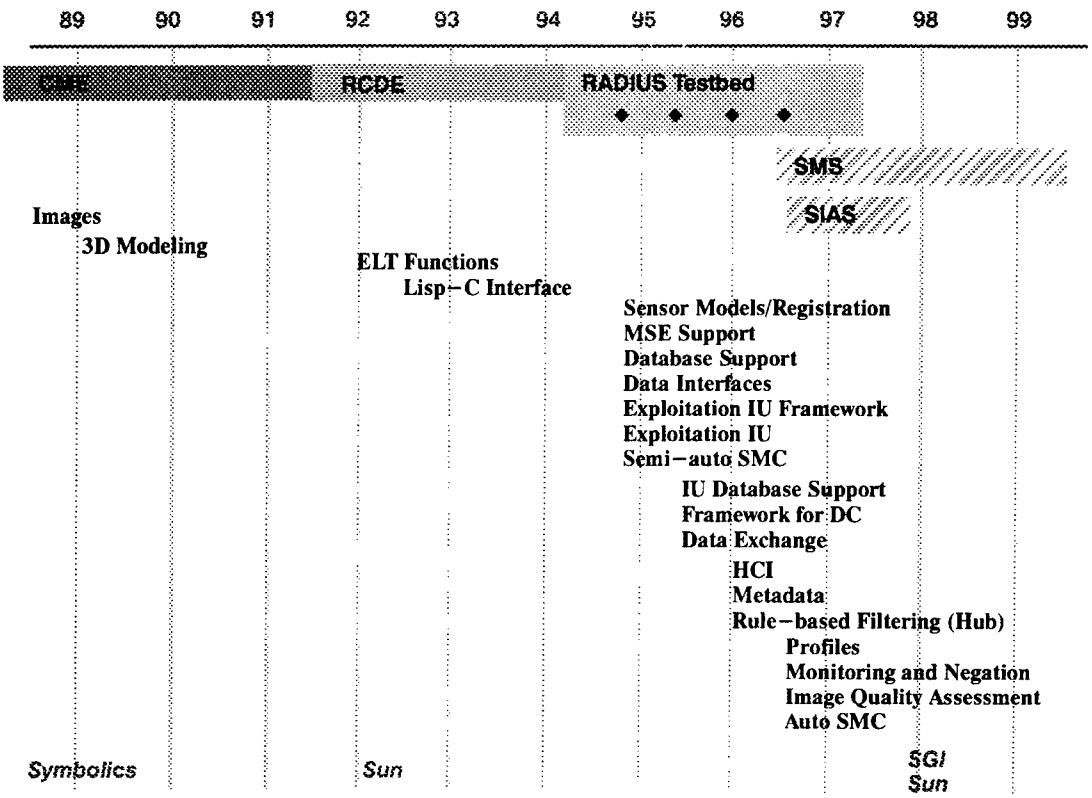


Figure 7: The evolution of the capabilities of the RADIUS Testbed.

cluded in the RTS, such as Khoros (UNM), TargetJr (GE) and Data Management Tool (LMC), but these subsystems support specific functions, such as particular IU algorithms, that are not part of the central processing core.

Implementing the MSE concept required a system that combines imagery, 3-D site models, and sensor geometry to display site models overlaid on imagery. Integrating IU technology created additional demands, such as support for development in C, C++, Lisp, and CLOS. Because the RCDE standardizes the way imagery and 3-D models are accessed, IU code development can be conducted at labs and universities, and the results easily integrated into the complete RTS.

Figure 7 shows the capabilities added to the RTS during Phase II. RADIUS used a spiral development methodology, in which multiple deliveries of the system were evaluated during the development period. The four diamonds in the RADIUS Testbed time-line represent incremental deliveries of the RTS to NIMA. Earlier deliveries, particularly the Initial Delivery, emphasized workstation capabilities such as image manipulation, manual object modeling, and database support. A large fraction of the RTS capabilities in the initial delivery were supplied by the RCDE, with topical modifications.

The major difficulty encountered in the Initial Delivery was in the externally provided data. Significant inconsistencies in the initial site models required an early development of bundle adjustment functionality to be added to the RTS, and related software applied to correcting the geometry of those models.

The initial delivery focused on producing a complete, prototype workstation as well as IU technology. The evolution of the IU capabilities of the RTS is shown in more detail in Figure 8. All of the algorithms in the initial delivery were enhanced in subsequent deliveries, as the

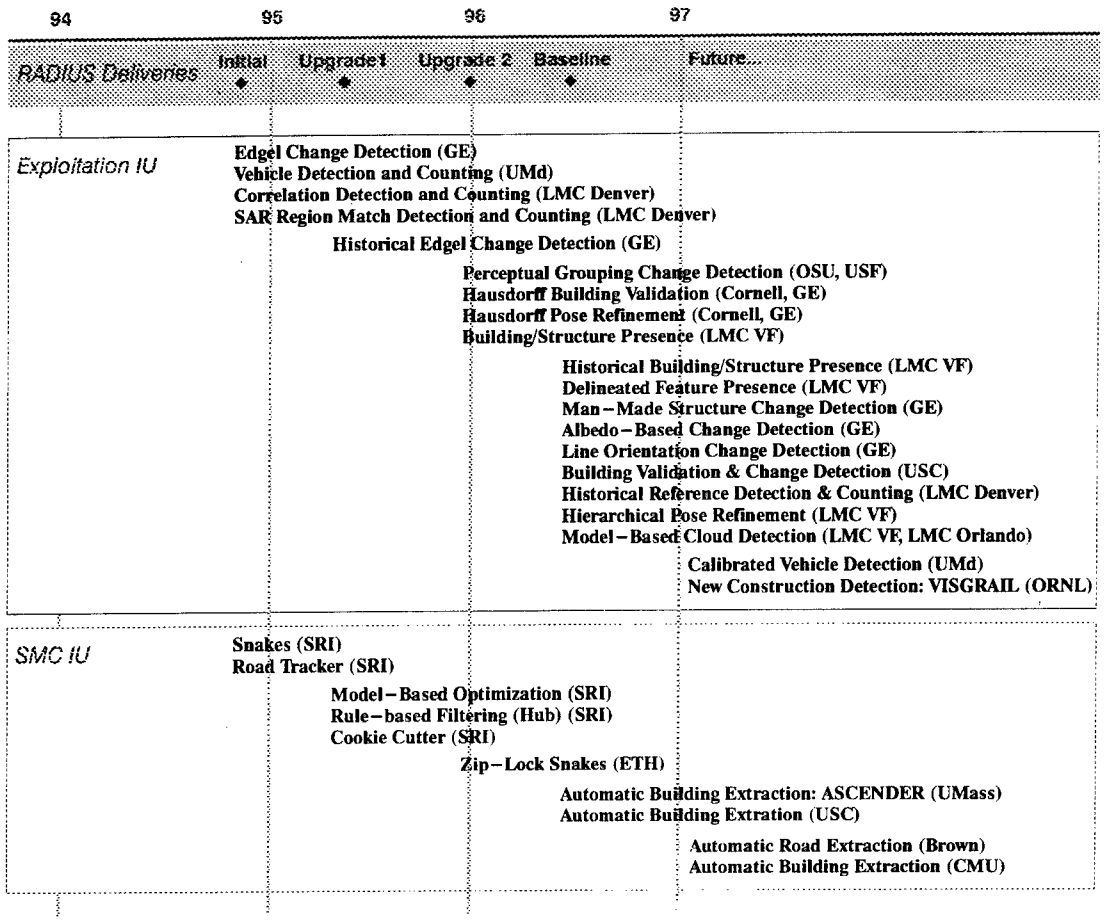


Figure 8: The evolution of RADIUS IU algorithms, as integrated into the RADIUS Testbed.

project emphasis shifted toward IU once the basic testbed capabilities were in place.

Exploitation and SMC algorithms are shown separately in the figure. While most SMC work was funded by other contracts, such as the DARPA RADIUS Broad Agency Announcement (BAA) contracts, development of Exploitation IU was largely funded by the RADIUS contract and its team members. Very little IU was available to be applied against exploitation tasks within the MSE framework. Consequently, the focus of the contract team members shifted toward developing Exploitation IU, while continuing to integrate site model construction systems. The figure shows the increase in the number of exploitation algorithms in later deliveries.

Originally, the project was planned to provide very specific workstation and IU capabilities at each delivery. After the first two deliveries, however, it became necessary to replace lower-priority requirements with unforeseen additions, such as data-oriented scenarios demonstrating the effectiveness of IU algorithms. This change was largely driven by a very successful RADIUS demonstration at the 1995 Exploitation Technology Symposium (ETS), an annual government-organized forum for discussing and demonstrating current efforts in automating imagery exploitation.

The focus of the last two deliveries was thereby adjusted toward task-based IU technologies solving real imagery exploitation problems. These deliveries included scenarios describing the use of particular algorithms to effectively perform important IA tasks on specific images. This delivery mechanism was effective, because it provided NIMA with demonstrations appro-

priate for a broad range of government personnel, and it helped algorithm developers focus on real problems of interest. One risk of this approach is that algorithms may become specialized to a narrow range of examples; to avoid this, scenario algorithms were tested on a number of images with varying conditions before being included in scenarios [Bremner et al., 1996].

At the end of the RADIUS Phase II contract, the RTS met over 90% of the contract specifications, comprised of requirements, goals and development guidelines. The system includes many capabilities not specified in the original plan, such as an extensive framework and user interface for exploitation IU, bundle adjustment, and support for automated registration through integration with the Model Supported Positioning (MSP) program. The suite of integrated algorithms is extensive, including support and complete user interfaces for: automated extraction of flat-roofed buildings; model optimization of all building types, linear features and area features; counting of generic vehicles in parking areas; counting of closely-parked vehicles in garrison areas; detection of change in fixed structures; detection of new construction; detection of the presence/absence of movable objects; model-based image quality assessment; and image registration refinement.

In addition, ATR capabilities are being integrated into the RTS and its descendents through the Model-Supported Target Recognition (MOSTAR) program. Automated site monitoring is being pursued in the Site Monitoring System (SMS) program, which is creating an operational prototype for tactical imagery analysis. The use of MSE is the focus of the Spatial Image Annotation System (SIAS), an operational prototype for national imagery analysis.

4 Major Accomplishments & Issues

As the RTS evolved to meet the specified requirements, goals, and guidelines, many derived capabilities were added as the need arose. The software development and integration efforts were designed to allow the various capabilities to be more than stand-alone tools. By maintaining consistency between top-down design and bottom-up implementation, the various components of the RTS operate synergistically to accomplish the major goals of RADIUS. The most significant achievements are briefly described below.

Adhering to the software development and integration standards that enable the cooperative interaction of the various RTS capabilities also had its problems. A few of the most significant impediments along the way are also described.

4.1 Accomplishments

RADIUS accomplishments can be sorted into four general categories:

- development of IU technology;
- development of the RTS;
- integration of IU algorithms into the testbed;
- laboratory evaluation in a nearly operational setting within NIMA.

IU Research The research conducted under the RADIUS program made significant contributions to the field of IU. RADIUS IU research focused on using context to improve algorithm performance – specifically, using the context available in the MSE framework. This approach allowed the development of simple, robust algorithms that operate successfully in a narrow context.

RADIUS also developed new methods of automated parameter adjustment. Because algorithms are executed on multiple images without human intervention, it is necessary for algorithm parameters to be adjusted based on observable image data (or not adjusted at all). RADIUS contains a number of such algorithms that offer three different methods of using historical imagery for calibration [Hoogs and Bajcsy, 1996, Mundy, 1996].

At a higher level, the RADIUS project demonstrated convincingly that IU researchers can work closely with IAs and NIMA personnel to

refine IU requirements. IU researchers were able to communicate effectively with IAs, other government personnel, and industrial experts to derive the necessary understanding of IA tasks to formulate useful IU algorithms.

Finally, the RADIUS program facilitated collaboration between a diverse group of researchers. IU technology was developed by three sources:

1. the RADIUS contractors;
2. the DARPA BAA contractors (University of Maryland, SRI, Carnegie Mellon, University of Massachusetts, University of Southern California, University of Washington);
3. other universities.

The last category includes institutions whose algorithms were reviewed by the Image Understanding Advisory Committee (IUAC). This committee was composed of members of the RADIUS Contractor team and members of the IU community, and was formed to conduct a survey of algorithms suitable for use in RADIUS.

The interactions within this large community of researchers stimulated research, collaboration and critical analysis, resulting in significant improvement in the overall quality of the IU in RADIUS.

Testbed Development The RTS prototype workstation has been critical to achieving three main goals of RADIUS. The existing RTS is:

- a near operational, prototype MSE workstation providing all the tools necessary to facilitate softcopy exploitation and report generation.
- a development and integration environment for IU researchers;
- a testbed for the evaluation of IU algorithms and for the exercise of various combinations of tools and interfaces;

IU Integration & Development The RTS evolved with the dual purpose of automating both feature extraction and exploitation. For exploitation IU, there is a documented integration protocol which was evolved during the contract. That protocol enables all exploitation IU algorithms to be easily incorporated into the standardized IU framework. This interface has enabled all of the current exploitation algorithms to be uniformly evaluated under operational conditions where many images may be applied with no user intervention.

While there is no integrating framework for site model construction algorithms, those algorithms that were integrated into the RCDE by the developers were transitioned into the RTS with minimal effort. The use of the RCDE as a common environment between IU developers and the RTS resulted in major savings in integration cost.

Automated Exploitation Processing The chain of processing for operational exploitation is completely automated in the RTS. This processing pipeline was accomplished with the integration of automated image-to-site-model registration from the MSP project. Images of the sites of interest are extracted from the available source imagery. That imagery is then registered using MSP, and added to the site. Then, image chips are analyzed using the automated IU Exploitation Framework and results stored for analysis.

This prototypical operational flow enables the IA to establish profiles to monitor features of interest for specific events. With a set of profiles created for a site, all new images of that site are processed automatically. The IA may examine the processing results, but no other effort is required.

Algorithm Evaluations Systematic, formal evaluations of IU algorithms frequently requires hundreds or thousands of images showing a variety of different imaging conditions. Since RADIUS did not have the resources to accomplish this level of evaluation, an alternate process evolved over the course of the project that allowed for reasonable evaluation while still providing feedback to developers. The careful se-

lection of a few images with "typical" imaging conditions was sufficient to determine the following: that the algorithm was performing as designed, that the algorithm was able to operate in its domain of applicability, and that the algorithm would be stable under at least a known range of conditions. It was not necessary (or possible, in some cases) for the developer to obtain a list of images with algorithm results. The benefit of this method was its ability to provide qualitative information to the developers, thereby enabling them to refine their algorithms. This process was used frequently and worked very well.

For site model construction, a standard set of buildings in specific sites were identified along with expected difficulties in each of several registered images. This data was used to perform comparative testing of similar SMC algorithms, and to validate that SMC software was performing as intended.

4.2 Issues

Multi-Purpose Testbed There were at least three separate areas of emphasis for the Testbed: as a research project, as a near-operational workstation prototype, and as a testbed for IU algorithm development and evaluation. As a research project, RADIUS was used to test the utility concept of MSE and various operational interfaces, data formats, and user interfaces. As an operational workstation prototype, RADIUS was expected to be relatively easy to use and stable under normal operation. As a testbed, RADIUS was a software development and integration environment for IU research for which both engineering and laboratory evaluations were performed.

In many situations, these areas of emphasis were in conflict. For example, the need to have a stable, robust, user-friendly workstation demanded that a great deal of effort be placed on HCI and testbed work. In addition, algorithms had to be routinely tested in all reasonable possible modes of operation. There was a clear trade-off between testing, evaluation, and testbed capabilities, and new algorithm development and integration.

The compromise reached on RADIUS was to strictly limit algorithm evaluation and formal system testing. Emphasis was placed on development and integration, with no formal testing procedure to ensure robustness or IU accuracy. This resulted in a large suite of capabilities, but a system that required more effort to operate by NIMA personnel.

Unanticipated Tasks There were numerous specific requirements which supported the several areas of emphasis for the RTS. It was also recognized that this was a spiraling development methodology, and, over the life of the project, there were regular re-plans of the directions of the contract. As might be expected, numerous unplanned tasks became necessary, requiring complex accommodations to incorporate them. The new tasks included:

- Development of site models without the use of detailed ground truth. A full bundle adjustment package for multi-image registration was developed to compensate for lack of ground truth in classified sites.
- Development of site models via a simple user interface. The process of creating site models was originally assigned to the contractors, and no user interface was developed. However, it became apparent that it would be useful to enable trained operators at NIMA to perform multi-image registration for site model initialization, and the RTS was enhanced with a user interface to this capability.
- Management of photogrammetry. Several studies on the accuracy of registration in the RTS were performed. These studies required the acquisition of ground truth and images, careful setup of experiments, as well as writing formal reports on the results.
- HCI. Originally, little emphasis was placed on user interfaces. As the RTS grew, however, it became clear that significant user interface enhancements were needed before the system could be presented to IAs. This

was particularly true of the user interface to exploitation IU systems. A significant portion of the testbed effort was then redirected toward making the testbed easy to use by IAs.

5 Future Directions

The RTS is expected to have two alternative future directions: transitioning some subsets of its capabilities to become operational systems, and enhancements to the system as a testbed for IU research and algorithm evaluation.

5.1 Technology Transfer

The following two tools are examples of technology transfer from RADIUS in a limited context.

SIAS The Spatial Imagery Annotation System is a technology transfer of part of the RTS to generate annotated image products and perform rudimentary model supported exploitation. It uses 3-D site models to reduce the effort involved in updating annotations on new imagery since previous annotations are correctly registered in 3-D on all new images. It also facilitates rapid IA orientation to the site by providing registered site model overlays on imagery. Making the HCI intuitive, stable and robust is a significant part of this effort, and a contribution to RADIUS technology.

SMS The Site Monitoring System is a part of the Semi-Automated IMINT Processing (SAIP) Advanced Concept Technology Demonstration (ACTD). Its goal is to automate the process of detecting changes in fixed sites using various imagery sensors. This spinoff of RADIUS technology will harden the end-to-end exploitation chain and test the exploitation algorithms on much more imagery than previously attempted. The expected result is a greater understanding of the range and depth of model-supported IU technology, leading to more robust algorithm performance.

5.2 Recommendations for Future Technology Development

The following are recommended as extensions to RADIUS technology and the current RTS.

Algorithm Generalization The exploitation algorithms in RADIUS operate on a relatively small region of an image for a very specialized purpose, without human interaction. At a minimum, each algorithm accepts a set of common inputs, and returns a set of common outputs. In this framework, questions that arise include the information that is available for input to these algorithms, the information that should be returned as the result of the processing, the information that should be gathered to feed back to the algorithm developer, and the information that should be gathered to allow combinations of various results. To some extent, each of these aspects has been explored in RADIUS, but more work needs to be done before operational use.

Context IU algorithms should take advantage of *all* the contextual information available, including sensor information, site feature information, high-level functional descriptions of image areas and historical imagery. Under RADIUS, algorithms were developed to exploit all of these forms of context, but the IU community is still learning how algorithms may fully benefit from information beyond image pixels. One particularly powerful form of context, learning from historical imagery, was used by algorithms within RADIUS, but its potential is still largely unrealized.

Self-Calibration A form of learning from historical imagery, the idea of using site model context to enable self-calibration of IU algorithms has proved to be a powerful, fundamental RADIUS concept. The RADIUS system identified the need for self-calibration, and the IU community should be encouraged to pursue how such context can be used in other aspects of algorithm operation.

Algorithm Exclusion Another key RADIUS concept is to use known algorithm lim-

itations to control algorithm execution. The RTS can prevent algorithm execution when a given image region is obscured by clouds, occluded by other model objects, or shadowed. The Hub provides a framework for developer-supplied rules to be invoked, controlling algorithm execution based on measurable image and site conditions. Algorithm exclusion should be further explored to improve robustness of results.

Inferences Effort should be directed toward making higher-level decisions of intelligence interest based on multiple IU results. One possibility is to combine the results of redundant-but-different algorithms executed on one feature to increase results confidence. A second possibility is to combine the results of many algorithms on many regions to infer an activity. Another area is to combine information derived from multiple sources and sensors, including multi-sensor fusion and cross-sensor cuing within the MSE framework.

Exploitation IU Under RADIUS, a number of exploitation IU algorithms were developed, since existing algorithms were not equipped to take advantage of MSE context. There are many more possibilities for exploitation algorithms, however, that have not even been explored on RADIUS. The two areas where new research would be most efficacious are change detection and robustness. Existing algorithms should be extended, improved and evaluated. New algorithms should be investigated and evaluated as they are identified.

Site Model Construction Automated and semiautomated SMC functions must be considerably augmented before there will be a real time savings in their usage. The automated SMC algorithms should incorporate simultaneous multi-image confirmation of roof-top edges, to improve robustness. Further research should be conducted to build a reliable linear feature extraction system, since linear features are time-consuming to construct manually.

Algorithm Evaluation While RADIUS developed a significant number of algorithms, resources were not available for thorough evaluations of those algorithms by the developers or users. Evaluations can provide invaluable feedback to developers, and can be used to establish exclusion rules, or domains of applicability, that greatly increase overall system robustness. Future resources should be directed toward algorithm evaluation, using scientific practices that guarantee the collection of appropriate data and accurate interpretation of that data.

RTS Development The following are suggested for near-term RTS development:

- **IA Tasking Language** A semantically organized system for selecting IU algorithms for very specific purposes should be integrated into the RTS, to help analysts choose the proper algorithm to perform a given exploitation task. Algorithms should be named and specialized for particular detection, recognition, and counting tasks.

- **Data Gathering** The infrastructure for gathering evaluation data should be extended to handle all aspects of the laboratory evaluation process. Experimental data can be extremely valuable for algorithm developers as well as users, but all appropriate information must be collected simultaneously. Some data gathering was performed under RADIUS, but more effort should be expended now that significant infrastructure is already in place.

- **Multi–Tasking** Operational versions of the RTS will likely have many processors available for near-real time computation. Since the RTS exploitation framework separates execution of data-parallel algorithms, minor modifications to the RTS should be made to enable porting to parallel hardware, improving overall exploitation throughput.

6 Acknowledgements

The RADIUS project has spanned a number of years. In the past three years, our focus has been the RTS, and many people at a number

of organizations have contributed, often making great personal sacrifices to do so. We would like to acknowledge them here, and thank everyone who has been a part of RADIUS. Note: individuals are listed once, although many have served in multiple capacities.

DARPA Oscar Firschein, Tom Strat, Rand Waltzman

NIMA Bill Glatz

Consultants Sid Wood, Doug Climenson, Ken Unger, Jim Green, Ed Mikhail, Bill Attaya, Dave McKeown

RADIUS Review Board Tom Hay (TEC), Merle Biggin (DMA), Mike Gilbert (CIO), Steve Moore (DIA)

RADIUS IU Advisory Committee Eamon Barrett (LMC Sunnyvale), Kim Boyer (OSU), Dan Huttenlocher (Cornell), Avi Kak (Purdue), Lavene Kanal (LNK Co.), Jean Ponce (U. of IL.), Azriel Rosenfeld (UMd), Demitri Terzopoulos (U. of Toronto), Ed Zelnio (Wright Labs)

Universities and Labs Rama Chellappa, Phillippe Burlina and Qinfen Zheng (UMd); Bob Collins, Ed Riseman and Al Hanson (UMass); Ram Nevatia, Keith Price, Andres Huertas and Gerard Medioni (USC); Bob Haralick (U. of Washington); Chris McGlone (CMU); Reinhold Mann and Ron Lee (Oak Ridge National Labs); David Cooper and Meir Barzohar (Brown); Sudeep Sarkar (U. of S. Florida); Walter Neuenschwander (ETH); Nandhu Nandhakumar (UVA); and many graduate students who did much of the work.

BDM Jan Sargent, Mike Kelly, Judy Bailey

GE Joe Mundy, Rich Welty, Rupert Curwen, Patti Vrobel

LMC Denver Chris Debrunner, Ann Chou, Lance Gatrell

LMC Valley Forge Mark Horwedel, Bethany Kniffin, Mike Puscar, Ray Cardenas, Tony Canike, Evelyn Arroyo, Tom Barrett, John Goodson, Harry Rosenthal, Bob Jordan, Mark Thompson, Norris Heintzelman, Juan Soto, Rick Hyson.

SRI Lynn Quam, Aaron Heller, Pascal Fua, Chris Connolly

GDE Walt Mueller

Texas Instruments Richard Ely

We gratefully acknowledge our government sponsors, and corporations that have contributed funding, including Lockheed Martin Management and Data Systems, SRI International, and GE CR&D.

References

[Bailey et al., 1994] J. Bailey, M. Kelly and J. Sargent. Quick-Look: A New Way to Prioritize Imagery for Exploitation. *Proceedings of the ARPA IU Workshop*, Nov. 1994.

[Barzohar and Cooper, 1993] M. Barzohar and D. Cooper. Automatic Finding of Main Roads in Aerial Images by Using Geometric-Stochastic Models and Estimation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 459-464, June 1993.

[Bremner et al., 1996] B. Bremner, A. Hoogs and J. Mundy. Integration of Image Understanding Exploitation Algorithms in the RADIUS Testbed. *Proceedings of the ARPA IU Workshop*, Feb. 1996.

[Chellappa et al., 1996] R. Chellappa, X. Zhang, P. Burlina, C.L. Lin, Q. Zheng, L.S. Davis and A. Rosenfeld. An Integrated System for Site Model Supported Monitoring of Transportation Activities in Aerial Images. *Proceedings of the ARPA IU Workshop*, Feb. 1996.

[Collins et al., 1995] R. Collins, Y. Cheng, C. Jaynes, F. Stolle, X. Wang, A. Hanson, E.

Riseman. Site Model Acquisition and Extension from Aerial Images. *Proceedings of the International Conference on Computer Vision*, June 1995.

[Fua, 1996] P. Fua. Cartographic Applications of Model-Based Optimization. *Proceedings of the ARPA IU Workshop*, Feb. 1996.

[Gerson and Wood, 1994] D. Gerson and S. Wood. RADIUS Phase 2: The RADIUS Testbed System. *Proceedings of the ARPA IU Workshop*, Nov. 1994.

[Hoogs and Bajcsy, 1996] A. Hoogs and R. Bajcsy. Model-Based Learning of Segmentations. *Proceedings of the International Conference on Pattern Recognition*, Vienna, Austria, 1996.

[Hoogs and Bajcsy, 1995] A. Hoogs and R. Bajcsy. Using Scene Context to Model Segmentations. *Proceedings of the IEEE Workshop on Context-Based Vision*, Cambridge, MA, 1995.

[Hoogs and Kniffin, 1994] A. Hoogs and B. Kniffin. The RADIUS Testbed Database: Issues and Design. *Proceedings of the ARPA IU Workshop*, Nov. 1994.

[Huertas and Nevatia, 1996] A. Huertas and R. Nevatia. Detecting Changes in Aerial Views of Man-Made Structures. *Proceedings of the ARPA IU Workshop*, Feb. 1996.

[Huttenlocher, 1993] D. Huttenlocher, G. Klanderman and W. Rucklidge. Comparing Images Using the Hausdorff Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, March 1993.

[Kniffin and Hoogs, 1996] B. Kniffin. and A. Hoogs. Database Support for Exploitation Image Understanding. *Proceedings of the ARPA IU Workshop*, Feb. 1996.

[Lin and Nevatia, 1996] C. Lin and R. Nevatia. Building Detection and Description from Monocular Aerial Images. *Proceedings of the ARPA IU Workshop*, Feb. 1996.

[Mundy, 1996] J. Mundy. Observation Events: A Basis for Change Detection. *Proceedings of the SPIE Conference on Applied Imagery and Pattern Recognition: Tools and Techniques for Modeling and Simulation*, pp. 89-109, 1996.

[Mundy and Vrobel, 1994] J. Mundy and P. Vrobel. The Role of IU Technology in RADIUS Phase 2. *Proceedings of the ARPA IU Workshop*, Nov. 1994.

[McKeown and Roux, 1994] D. Mckeown and M. Roux. Feature Matching for Building Extraction from Multiple Views. *Proceedings of the ARPA IU Workshop*, Nov. 1994.

[Neuenschwander et al., 1994] W. Neuenschwander, P. Fua, G. Szekely, and O. Kubler. Initializing Snakes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 658-663, June 1994.

[Sarkar and Boyer, 1993] S. Sarkar and K. Boyer. Integration, Inference and Management of Spatial Information Using Bayesian Networks: Perceptual Organization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, March 1993.

RADIUS Technology Transfer

Donald J. Gerson
ORD
Box 4132
Washington, D.C. 20505
dgerson@snap.org

Sidney E. Wood
SAIC/ORD SETA
Box 4132
Washington, DC 20505
swood@snap.org

William R. Glatz
NIMA
Building 213, WNY
Washington, D.C. 20505
bglatz@snap.org

ABSTRACT

Research and Development for Image Understanding Systems (RADIUS), a two-phase five-year project, is aimed at increasing Imagery Analyst (IA) productivity, and improving the quality and timeliness of IA products. A key feature of RADIUS is Model-Supported Exploitation (MSE) in which two-dimensional and three-dimensional models of a site are used as the foundation for subsequent analysis and reporting. Image understanding (IU) technology is an integral part of this support to analysis. The RADIUS Project is now complete, and action is being taken to transfer selected technology to users. This paper describes the challenges and opportunities encountered in the transfer process. The available capabilities, existing shortcomings, and the improvements to meet the real-world needs of imagery exploitation are described.

Keywords: imagery analysis, imagery interpretation, imagery exploitation, image understanding, model-supported exploitation, site model, automatic target recognition, assisted target recognition.

1. Introduction

RADIUS was a joint project of the Central Intelligence Agency (CIA) and the Defense Advanced Research Projects Agency (DARPA) aimed at increasing IA productivity and improving the quality and timeliness of their products. The approach for meeting this challenge was based on the application of advanced technology to the imagery analysis process. The fundamental concept of RADIUS was to provide salient information to IAs directly, using IU algorithms that operate as "intelligent assistants" to IAs.

1.1 The Site Model Concept

Information to aid the imagery analysis process is provided through three-dimensional models of the site being analyzed, in several ways. The most obvious of these is through the use of 3-D wire frame depictions of the structures, roads, and other natural and man-made features and/or objects at a site. This model provides information on the physical characteristics of the features such as shape, size, position, orientation, and interrelationships of the features. Another source of information is provided through demarcating and labeling specific regions, called "functional areas," or of the specific buildings, roads, etc. This capability can provide information *about* these objects through point and click. Extensive additional information about these site features could be made available. Some of these include: use, history, composition, and relevance; or information on the activities to expect in various functional areas within the site, and the potential importance of those entities or activities. Labeling may also be extended to provide access to other collateral information concerning the site that has not been derived from imagery. The source of this other information could include a variety of databases. This entire concept is called "Model-Supported Exploitation" [Gerson and Wood, 1994]. A potentially extremely powerful, but currently immature, concept is called the "Hub" [Strat, Fua and Connolly, 1997]. The Hub automatically selects the appropriate algorithms, and/or the parameters used by the algorithms, based upon the characteristics of the site, image acquisition parameters and conditions, and the design capabilities of the algorithms. At present, Hub is used only within one semiautomatic site model construction algorithm, where it sets parameters based on the object being modeled and imaging conditions. This "trained" selection process will enable more fully automatic and robust computer processing.

1.2 Operational Requirements

RADIUS was designed to support IAs dealing with real-world operational intelligence problems. We are well into an era in which sensors are providing orders of magnitude increases in quantity, quality, and diversity of imagery. At the same

time, the Intelligence Community is experiencing a significant reduction in personnel, including IAs. Consequently, technology such as that developed within RADIUS which show high potential for increasing IA productivity, are important efforts to provide the Imagery Intelligence (IMINT) support required at national, theater, and tactical levels. The focus of RADIUS technology transfer is to begin to meet the needs of operational organizations.

1.3 Imagery Exploitation Tasks

The goal of imagery exploitation in RADIUS was to use imagery from a variety of sensors. With extensive input from a wide variety of IAs, a set of image exploitation tasks, with the potential of being supported by IU, was defined prior to the beginning of RADIUS, some of which were eliminated or modified as the project developed and needs and capabilities became better understood.

The general Detection and Counting Task is closely related to Automatic Target Recognition (ATR) and was pursued in the MOdel-Supported TArget Recognition (MOSTAR) Project [Allmen, et al., 1996], a joint National Reconnaissance Office (NRO)/DARPA/CIA project. Detection and Counting was pursued in RADIUS as part of site monitoring, but the MOSTAR effort went further, and explored the use of traditional ATR technology. The Model to image and image to image Registration Task was investigated in the Model-Supported Positioning (MSP) Project [Mueller, 1997] and [Ely and Di Girolamo]. This critical task was not pursued under RADIUS directly, but is absolutely necessary for accomplishment of most IU support within the MSE concept.

The key exploitation technology made possible by the RADIUS 3-D site model is the automatic detection of change at a site, at which the desired changes to be monitored are pre-defined in a process called First Look. This approach is discussed briefly in Section 1.5.2, and more completely in [Hoogs and Hackett, 1994], [Chellappa, et al., 1997], and [Gee and Newman, 1993]. This technology made it possible for RADIUS to pursue the Site Monitoring, Negation, and Trends and History applications. Note that the focused Change Detection Task differs from the broader human task to "detect, identify, or forecast all changes of intelligence significance at a given site," which is not possible by fully automated means within the present state of the art in IU and related technologies.

1.4 Model-Supported Exploitation

MSE embodies the concepts that were addressed in Section 1.1. The most important of the many technologies needed to develop automated MSE capabilities are site model construction, image-to-model and image-to-image registration, database utilization, and Change Detection. IU, the major field of research in the RADIUS Project, can be used in each of these technology areas. While interactive applications of IU are relatively new, the operational use of such techniques was a major area of study in the project. These technologies are described briefly in the paragraphs that follow.

Site Model Construction: Site models consist of two- and three-dimensional geometric descriptions of fixed features at the site, along with supporting collateral information about the site and source data. Site models provide a common geographic reference for any information about the sites. Site models can be constructed without the aid of IU, but the work is lengthy and tedious, and one major thrust of RADIUS was to sponsor research and to develop IU approaches that will be of benefit to the site model construction process.

Registration: As stated above, registration functions were addressed separately from the RADIUS Project. The capability to precisely register a new image to a site model automatically is key to the entire RADIUS concept, and the results of the separate MSP registration contract were incorporated into the RADIUS Testbed System (RTS).

Database Support: Further database technology development is required for several MSE applications. The primary requirement is to make collateral data available, via the site models, through the point and click technique. A vast amount of current and historical data, of greatly varying formats and types, must be accessible through this technique. There is also a requirement for databases of the site models themselves, and of current and historical imagery. All of these must interface with other, separate intelligence and operational databases.

Change Detection: In the RADIUS context, Change Detection means the detection of changes of intelligence importance at a particular site and at a particular time. In many cases, the types of changes are predefined, e.g., "Has the number of vehicles in this Motor Pool changed by more than 20 percent since last imaged?" These predefined Change Detection "triggers" can typically be associated with specific locations, or "functional areas" within a site, one of the benefits of MSE. Change Detection is a classic IU application that was demonstrated in RADIUS Phase II.

1.5 The Value of Site Models

The site model plays a key role for IA visualization purposes and for the use of IU algorithms as

the basis for automated or IA-assisted site monitoring, as described below.

1.5.1 Site Models Used for Visualization

The availability of 3-D site models on a soft copy workstation will enable the production of site graphics in soft-copy or hard-copy form. These models can be provided to intelligence and operational users for visualization from any perspective, distance, altitude or aspect, or even in "fly-through" or "drive-through" form, including ingress and egress routes. The users will then be able to use the graphics, in conjunction with collateral information, to extend the knowledge base about the site. The RADIUS site model enables complete visualization of the site for planning purposes or for conduct of operations. For example, one might consider the value of site models to support the rescue of non-combatants in a crisis situation, in military operations in wartime, or in humanitarian and peace-keeping operations.

1.5.2 Site Models Used for IU Algorithms

The site model is helpful for image exploitation in general, and specifically for the First Look process. It is used for two purposes: (1) to focus the attention of the IU algorithm on specific regions of the site, and (2) to store information of use to the algorithm related to these selected regions. For example, roads or parking areas can be identified in the site model for use in detecting and counting vehicles. When a new image of the site becomes available, the image is registered to the site model, and the IU algorithm can examine the indicated roads and parking areas and display its findings. In addition, the site model can provide contextual information that aids in the selection of the proper IU algorithm or for the automatic adjustments of selected parameters affecting its performance.

1.5.3 Site Model Construction

Site models can be built manually, but at a high cost in terms of time, money, and personnel. Because the availability of site models was crucial to the success of RADIUS, a major effort of the project was to make the site model construction and updating process more efficient. Papers by [Chellappa, et al., 1997], [Collins, et al., 1997], [Fua, 1997], [Lin and Nevatia, 1997], [Hseih 1996], and [Noronha and Nevatia, 1997] describe the site model building effort that was conducted by various research organizations using both semiautomated and fully automated IU techniques.

2. RADIUS Phase I

RADIUS was a two-phase, five-year project. The two-year first phase included Concept Definition, advancement of the RADIUS Common Development Environment (RCDE), and initiation of six DARPA-sponsored RADIUS Research contracts related to various RADIUS requirements. During the three-year second phase, the RTS was developed and incrementally improved. The RTS became the focus for integration of results produced throughout this expanded RADIUS Community.

2.1 Concept Definition

Phase I Concept Definition [Edwards 1992, Gee 1993] centered on defining the Phase II RTS Requirements, the RTS Operations Concept (OPSCON), and the Preliminary RTS Evolution Plan. The major inputs to these documents were derived from four Concept Validation Experiments (CVEs) conducted over the course of Phase I [BDM 1993]. The CVEs resulted in a better understanding of (1) the features, objects, and collateral information which should be included in site models and accompanying Site Folders, (2) how site models might be used to support Change Detection and other exploitation tasks, (3) whether the payoff of MSE would be worth the investment of time and effort required to build and maintain site models, and (4) the kinds of human-machine collaboration that are appropriate for IU processing. A RADIUS Testbed System Architecture and Functional Design was also produced during this period.

2.2 The RADIUS Common Development Environment

A separate contract led to development and testing of the RCDE, an IU development environment tailored to the needs of the RADIUS MSE concept [Mundy 1992]. The RCDE facilitates the transfer and rapid integration and testing of technology, developed at various organizations, into the evolving RTS. During the RTS Phase, the contract team and the DARPA-sponsored RADIUS Research contractors used RCDE to receive and test the developed technologies.

2.3 RADIUS Research Contracts

DARPA sponsored a set of research and development studies in support of RADIUS. The DARPA contractors addressed Automated Cartographic Feature Extraction for Site Modeling (Carnegie Mellon University) [Hseih, 1996], Model-Based Optimization Approach to MSE (SRI International) [Fua 1997], Site Model-Based Image Registration and Change Detection (University of Maryland) [Chellappa 1997], Automated Site Model Acquisition and Extension (University of Massachusetts) [Collins 1997], Site Model-Based Change Detection (University of Southern California) [Lin 1997, Noronha 1997], and Perform-

ance Characterization of Computer Vision Algorithms (University of Washington) [Haralick 1997].

These contracts continued well into the RTS Phase, with close cooperation and coordination between and among the Government sponsors, the RADIUS Research contractors, and the RTS and MSP contract teams. The primary fora for broader IU Community (IUC) participation in RADIUS were semiannual RADIUS Development Workshops (RDWs), in which all of the aforementioned participated, and to which others in the academic and industry IUC were encouraged strongly to become active participants. These meetings were designed with two purposes in mind: to inform the community of RADIUS progress, problems, and plans; and to call upon its members to contribute, through presentations of proposed solutions to the problems identified.

3. RADIUS Phase II

The three-year RADIUS Phase II, or RTS Phase, commenced in March 1994. The RTS Phase included the development of a dual RTS, a Development Testbed (DTB) at the prime contractor's facility, and an Evaluation Testbed (ETB) installed at the Government site. This Phase of RADIUS was a model building, MSE, IU software Research and Development (R&D) project, some of the results of which were transferred to operational imagery exploitation workstations as technology transfer spin-offs during the project. RADIUS technologies are available to development organizations for incorporating selected advanced techniques for operational systems.

As stated in Section 1.5, site model building and updating and IU-supported MSE were the underlying concepts developed, demonstrated, tested, and evaluated on the RTS. The hardware suite in Phase II was simply the means through which the government tested and evaluated the MSE IU algorithms brought to the platform by the participants. In addition to the RTS, these included the MOSTAR and MSP contract teams, the DARPA RADIUS Research contractors, and other members of the academic and industry IUC who proposed and provided their algorithms as solutions for the various imagery exploitation tasks undertaken by RADIUS.

3.1 The RADIUS Testbed System

The RTS Phase called for the contract team to design a system that could be modified easily, install a core MSE end-to-end capability as quickly as possible, conduct tests and evaluations of IU algorithms designed to meet the requirements of the IA tasks discussed previously, and introduce incremental upgrades and improvements based on the results of the tests and evaluations. The RTS contract team acquired, assessed, adapted, installed, tested, and evaluated IU technologies on the DTB, at the prime contractor's facility, in preparation for installation and further test and evaluation on the ETB. Emphasis was placed on the acquisition of available IU technology from industry and academia. Development of new IU technology was undertaken by the RTS contract team only if determined that the needed technology was unavailable elsewhere. As the role of IU in MSE became better understood, and promising IU technology was available from outside sources, it was evaluated by the RTS contract team for possible adaptation, enhancement, and integration into the RTS.

3.2 Technology Transfer

The contract team was charged to design, integrate, install, and test Initial Delivery versions of the DTB and ETB nine months into the contract. Initial Delivery was accomplished in October 1994, with subsequent Upgrade Deliveries in March 1995 and December 1995. The Baseline Delivery was accomplished in July 1996. This process included delivery of a database of Government-provided site models and Site Folders of real-world sites of intelligence interest.

The RTS contract team assisted in the transfer of proven technology from the ETB to operational systems. A Final Report generated by the contract team provided full documentation of RADIUS capabilities, and addressed the feasibility of RADIUS technology transfer to meet user requirements. Two technology transfer projects are already underway. The Spatial Image Annotation System (SIAS), a RADIUS component, will be installed for operational testing at three sites in the spring of 1997. The RTS will serve as the basis for the Site Monitoring System (SMS) in the DARPA-sponsored Semi-Automated IMINT Processing (SAIP) Advanced Concept Technology Demonstration (ACTD) in 1997 and 1998.

4. RADIUS Transition Plans

RADIUS Phase II was completed at the end of March 1997, with the delivery of the RTS hardware and software, with its set of IU algorithms and other MSE and Electronic Light Table (ELT) capabilities, a Final Report with conclusions and recommendations for future development, and a Users Manual. During the final year of Phase II, RADIUS-developed capabilities were evaluated by the operational community and several components were considered mature enough to be transitioned into operational imagery exploitation environments.

4.1 Current Status

SIAS, a method of easily and accurately registering IA annotations from a previously exploited image to a new incoming image, is currently being installed for operational evaluation in three operational facilities. SIAS accomplishes registration of annotations by the use of a three-dimensional "annotation model," actually a mini-site model. There has already been interest in expansion of the annotation model to include footprints of objects such as buildings, which are normally found in full site models. This interest may lead to a greater proliferation of more complete site models in the operational arena.

SMS, based on the RTS, is currently being integrated as the Site Monitoring System for the Enhanced Delivery Phase of the SAIP ACTD to take place in November 1977. The SMS is a streamlined version of the RTS, installed on a multiprocessor Silicon Graphics (SGI) workstation, with a selected set of RTS EO and SAR IU algorithms suitable for the site monitoring task. The selected components of MOSTAR, MSP, and the Assisted Target Monitoring System (ATMS) will be incorporated to provide additional capabilities.

4.2 Transition to NIMA

The National Imagery and Mapping Agency (NIMA) was recently established as the DoD Combat Support agency with overall responsibility for coordination and execution of imagery exploitation R&D. With the 1997 completion of RADIUS and its associated projects; MOSTAR, MSP, SIAS, and SMS; plans are now in motion to transition some capabilities of each of these projects to NIMA for further research, refinement, evaluation, and/or selective technology transfer into the operational environment. The government management of the SMS Project, which will continue into 1998, will also transfer to the NIMA Research, Development, Test, and Evaluation (SR) organization in October 1997.

At the time of this writing, the NIMA long-term program for continuation of IU and ATR automated and assisted image examination research and development is not finalized. Initial plans are that the existing RTS will be maintained and upgraded for hosting on either or both the UltraSparc2 system developed for SIAS, or the SGI system developed for SAIP/SMS. Since neither of these systems contain full RTS capability, resources will be required and expended to enhance and integrate the missing components. Plans have been made for more extensive testing of algorithms and tools potentially planned for integration into operational systems. The RADIUS project did some characterization and limited evaluation of integrated algorithms, but no exhaustive testing of the kind required for operational consideration has yet been conducted. In addition, NIMA has interest in the research being conducted under the new DARPA Image Understanding for Battlefield Awareness (IUBA) Program. A number of the research projects included in that program are direct extensions and improvements of the earlier RADIUS Research activity. It may be useful and desirable to integrate some of these new research efforts into the expanding NIMA testbed activities.

References

M. Allmen, C. Debrunner, R. Hamilton, J. Layne, and R. Bondeson, "Integrated Use of 3D Site Models and ATR in SAR Image Exploitation," ATRWG System and Technology Symposium, Laurel, MD, July, 1996.

R. Chellappa, Q. Zheng, C. Shekhar, S. Kuttikkad, and P. Burlina, " Site Model Construction and Positioning Techniques for the Exploitation of Infrared and SAR Images," Chapter 3, RADIUS: Image Understanding for Intelligence Imagery, Oscar Firschein and Thomas M. Strat (eds.), Morgan Kaufman Publishers, San Francisco, CA 94104.

BDM, "Report on the RADIUS Concept Definition Experiments," August 20, 1993, 1501 BDM Way, McLean, VA 22102.

R. Chellappa, P. Burlina, X. Zhang, C.L.Lin Q. Zheng, L.S.Davis, and A. Rosenfeld, Site Model Mediated Detection of Movable Object Activities," Chapter 4, RADIUS: Image Understanding for Intelligence Imagery, Oscar Firschein and Thomas M. Strat (eds.), Morgan Kaufman Publishers, San Francisco, CA 94104.

R. Collins, C. Jaynes, Y-Q Cheng, X. Wang, F. Stolle, H. Schultz, A. Hanson, and E. Riseman, "Automatic Construction of Three-Dimensional Buildings," Chapter 3, RADIUS: Image Understanding for Intelligence Imagery, Oscar Firschein and Thomas M. Strat (eds.), Morgan Kaufman Publishers, San Francisco, CA 94104.

J. Edwards, S. Gee, A. Newman, R. Onishi, A. Parks, M. Sleeth, and F. Vilnrotter, "RADIUS: Research and Development for Image Understanding Systems - Phase I," Proceedings DARPA Image Understanding Workshop, San Diego, CA, January 1992, Morgan Kaufmann Publishers, San Mateo, CA 94403.

R. Ely and J. Di Girolamo "Automated Model-to-Image Registration," Chapter 2, RADIUS: Image Understanding for Intelligence Imagery, Oscar Firschein and Thomas M. Strat (eds.), Morgan Kaufman Publishers, San Francisco, CA 94104.

P. Fua, "Model-Based Optimization: An approach to Fast, Accurate, and Consistent Site Modeling from Imagery," Chapter 3, RADIUS: Image Understanding for Intelligence Imagery, Oscar Firschein and Thomas M. Strat (eds.), Morgan Kaufman Publishers, San Francisco, CA 94104.

S. Gee, and A. Newman, "Automating Image Analysis Through Model-Supported Exploitation, "Proceedings ARPA Image Understanding Workshop, Washington, DC, Apr. 1993, Morgan Kaufmann Publishers, San Mateo, CA 94403.

D. Gerson, and S. Wood, "RADIUS Phase II - The RADIUS Testbed System," Proceedings DARPA Image Understanding Workshop, Monterey, CA, November 1994, Morgan Kaufmann Publishers, San Mateo, CA 94403.

R. Haralick, "Performance Characterization of Computer Vision Algorithms," Chapter 7, RADIUS: Image Understanding for Intelligence Imagery, Oscar Firschein and Thomas M. Strat (eds.), Morgan Kaufman Publishers, San Francisco, CA 94104.

Y. Hseih, Design and Evaluation of a Semi-Automated Site Modeling System," Proceedings DARPA Image Understanding Workshop, Palm Springs, CA, February 1996, Morgan Kaufmann Publishers, San Mateo, CA 94403.

A. Hoogs and D. Hackett, "Model-Supported Exploitation as a Framework for Image Understanding," Proceedings DARPA Image Understanding Workshop, Monterey, CA, November 1994, Morgan Kaufmann Publishers, San Mateo, CA 94403.

C. Lin and R. Nevatia, "Building Detection and Description from a Monocular Image," Chapter 3, RADIUS: Image Understanding for Intelligence Imagery, Oscar Firschein and Thomas M. Strat (eds.), Morgan Kaufman Publishers, San Francisco, CA 94104.

W. Mueller and "Geometric Refinement in Model Supported Positioning," Chapter 2, RADIUS: Image Understanding for Intelligence Imagery, Oscar Firschein and Thomas M. Strat (eds.), Morgan Kaufman Publishers, San Francisco, CA 94104.

J. L. Mundy, R. Welty, L. Quam, T. Strat, W. Bremner, M. Horwedel, D. Hackett, and A. Hughes, "The RADIUS Common Development Environment," Proceedings DARPA Image Understanding Workshop, San Diego, CA, January 1992, Morgan Kaufmann Publishers, San Mateo, CA 94403.

S. Noronha and R. Nevatia, "Detection and Description of Buildings from Multiple Aerial Images," Chapter 3 RADIUS: Image Understanding for Intelligence Imagery, Oscar Firschein and Thomas M. Strat (eds.), Morgan Kaufman Publishers, San Francisco, CA 94104.

T. Strat, P. Fua and C. Connolly, "Context Based Vision," Chapter 5 RADIUS: Image Understanding for Intelligence Imagery, Oscar Firschein and Thomas M. Strat (eds.), Morgan Kaufman Publishers, San Francisco, CA 94104.

The views expressed in this manuscript are those of the authors, and not necessarily those of ORD or NIMA

FOCUS: A Shared Vision Technology Transfer Project

Eamon B. Barrett and Paul M. Payton
Lockheed Martin Missiles and Space Company
Org. D3-10, Bldg. 195D
1111 Lockheed Martin Way, Sunnyvale CA94089
E-mail: eamon.barrett@lmco.com
E-mail: payton@dipl.rdd.lmsc.lockheed.com

Joseph L. Mundy
General Electric Corporate Research and Development Laboratories
Box 8, Schenectady, NY12309
E-mail: mundy@crd.ge.com

Abstract

FOCUS is an ongoing "shared vision" (collaborative) IR&D project, jointly sponsored by Lockheed Martin Missiles and Space (LMMS/Sunnyvale) and General Electric Corporate Research & Development Laboratories (GE CR&D/Schenectady). The technical thrust of FOCUS is content-based retrieval and prioritization of data from massive imagery archives and data streams; the ultimate objective of FOCUS is to dramatically enhance the capability of image analysts (IAs) to cope with their high-volume workload. The approach we are taking with FOCUS is to tranfer Model Supported Exploitation and Image Understanding technology, developed under the DARPA/ORD RADIUS Program (and related Government-sponsored programs), and to adapt this technology for automatic off-line examination of archived images and image streams.

1 Introduction

In this paper we describe an ongoing R&D project conducted by LMMS and GE CR&D. We have named this project "FOCUS". FOCUS accomplishes automated change detection and cueing (ACDC) and prioritization of imagery as a precursor to interactive exploitation by IAs; FOCUS may also be viewed as a tool for content-based imagery retrieval (CBIR) from archives. FOCUS is intended to support: (a) the day-to-day monitoring and detection of significant changes and events in a specific geographic area; for example, an urban or industrial complex or a battlefield, and (b) IAs who have to contend with more images and data than they have resources to assimilate. FOCUS will help its users reduce and prioritize such workloads. FOCUS supports queries which specify the high-priority sub-areas and changes of interest at a site. This is accomplished by reference to high-resolution maps and/or models and reference images of the regions being monitored. As new images are collected, FOCUS automatically registers them to pre-existing or user-developed site models or site maps, and runs pixel-level algorithms which automatically detect specified changes from normalcy in the selected sub-areas. The results are presented as a prioritized list of small image chips, enabling the user to select at a glance a manageable subset of high-payoff data for interactive review. These "chips" are assembled into an analyst "tip sheet" to facilitate rapid image review prior to online image exploitation.

Lockheed Martin and General Electric initiated the FOCUS project in January 1996. MSE and IU technology development for

FOCUS is conducted by Dr. Joseph L. Mundy's group at GE CR&D, using TargetJr as the MSE/IU platform. Funding for this FOCUS technology development at GE CR&D is provided by the Lockheed Martin Corporation. A major objective of the FOCUS project is to transition MSE and IU research results into IDEX II, a fielded and proven image archival and dissemination system. At the IDEX Development Laboratories in Sunnyvale, Dr. Eamon B. Barrett and Mr. Paul M. Payton are interfacing these content-based image retrieval mechanisms to the IDEX imagery archive, under IR&D support provided by the Information and Computing Technologies Directorate of the Lockheed Martin Advanced Technology Center (ATC, formerly the Palo Alto Research Laboratories). Our exploratory development in 1996 was confined to panchromatic imagery, and initial results on unclassified and classified data were successfully demonstrated at the IDEX Users Forum, held in Sunnyvale on October 23-24, 1996. In 1997 the range of FOCUS applications is expanding to include Space Imaging International (SII) and other multispectral imagery, since IDEX is evolving in the direction of full commercial remote-sensing data archival capabilities

2 FOCUS Concept of Operations

Reconnaissance and commercial remote sensing imagery will generally be accompanied by "metadata", such as:

- collection system geometry
>> camera location
>> roll/pitch/yaw
>> focal length
>> principal point

- intended target region
>> corner coordinates
>> WAC/ONC cell
>> center point
>> target Ids

- acquisition date/time
>> GMT of image
>> flight number
>> strip/pass number

- image size
>> line/samples/bands
>> bits/pixel
>> spectral band assignment

- processing history
>> calibrations applied
>> radiometric enhancements
>> contrast adjustments
>> lookup and mapping tables
>> tonal transfer curves

This metadata provides a very valuable but coarse-grain query mechanism for initial screening of the image stream and retrieval of images from an archive. The FOCUS concept adds a second level of fine-grain filtering based on automated examination of image internals, such as pixel-level indications of changes from reference normalcy in user-specified geospatial locations. The two-level metadata-plus-content query mechanism is depicted schematically in Figure 1.

In our Sunnyvale IDEX Development Laboratory, FOCUS model registration and change detection algorithms run on a UNIX platform which is interfaced to IDEX through an Output Data Server (ODS). This interface provides FOCUS access to a large archive of NTM imagery in a development environment for an existing fielded system. The ODS also provides basic image manipulation functions, such as image retrieval from the IDEX image cache and format conversion from the IDEX compressed imagery (TFRD) to TIFF and, soon, NITF 2.0 formats acceptable to the FOCUS processor. Our work with the FOCUS prototype identifies requirements for future operational systems, such as extension of the image archive to incorporate site maps and 3D models in vector and raster formats as needed to support CBIR queries. The concept of incorporating content-based queries into the IDEX architecture is illustrated schematically in Figure 2, which shows the FOCUS tasking and query station as separate from the IA's imagery exploitation workstation (IES). In this configuration, once the content-based query mechanism is in place, the IA may commence the exploitation session by using the FOCUS-derived "tip sheet" to review the prioritized order of the task packet of images which are about to be exploited.

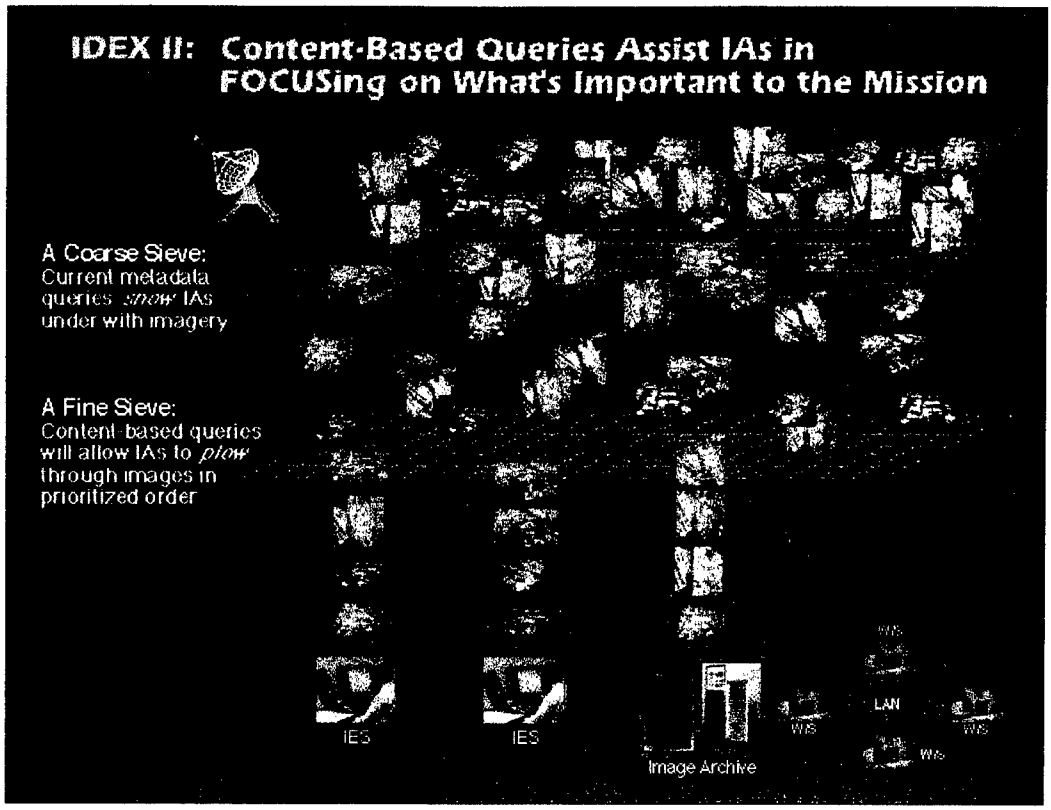


Figure 1. The FOCUS concept: Content-based queries are powerful image analyst tools

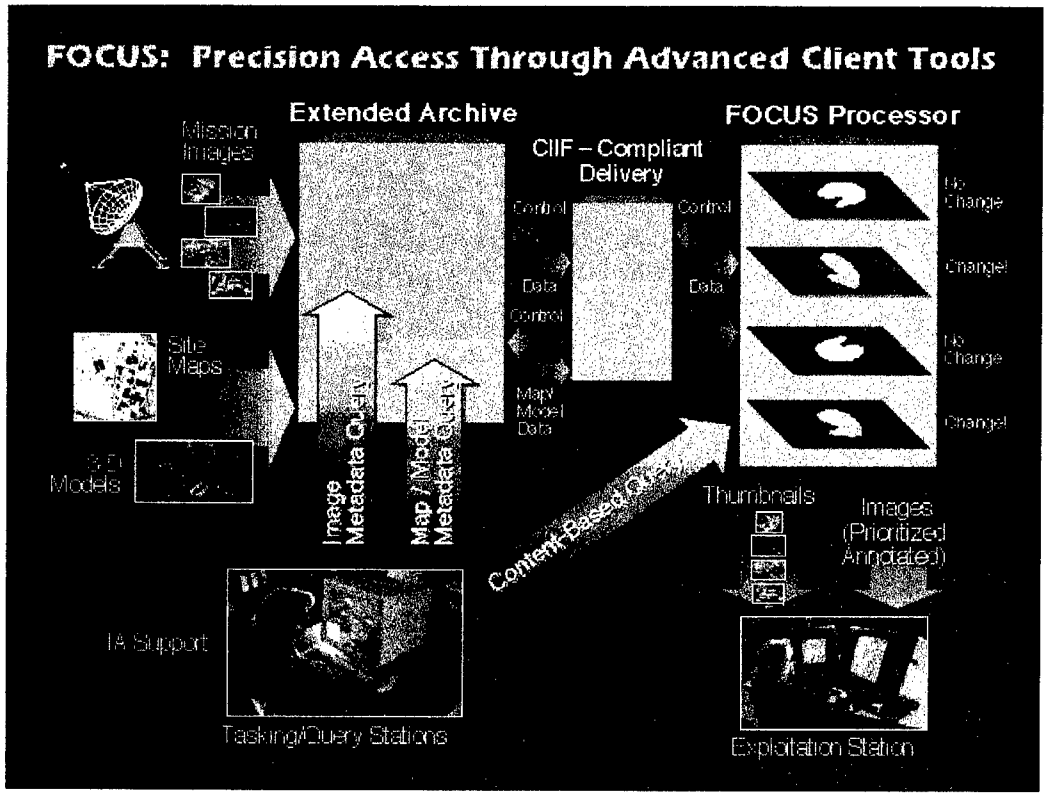


Figure 2. A block diagram of the FOCUS operational concept -- how content-based queries fit into the existing IDEX architecture

3 Examples of Automated Change Detection, Content-Based Retrieval, and Prioritization of Imagery by FOCUS

Our experimental work in the IDEX Development Laboratory utilizes NTM data to develop image understanding capabilities required by the special characteristics of these sensor types. We are precluded from publishing such data in these proceedings. In order to illustrate the concepts for this audience, we have also exercised FOCUS on unclassified aerial images, maps, and 3D models of our Lockheed Martin Missiles and Space home facility adjacent to Moffett Field in Sunnyvale, California.

This data set is described in our web site.
`http://badger.parl.com/~payton/iu/.`

The FOCUS process begins with metadata-based queries to the archive. The metadata for our unclassified aerial imagery database includes the approximate date of acquisition and the approximate latitude and longitude coordinates of the corner points of each image. Figure 3 shows the footprints of these images plotted in registration to a TIGER map of the region.

The metadata criterion selects the images which will be considered for FOCUS processing; we wish to further restrict this list by imposing a content-based criterion. This criterion could support analysts who must monitor a facility constantly or might prioritize a set of images which must be exploited in a limited period of time. Examples of a content criterion are 'retrieve only those images with an aircraft parked in a specific spot' or 'retrieve only those images where a building was/was not observed in this geographic location'.

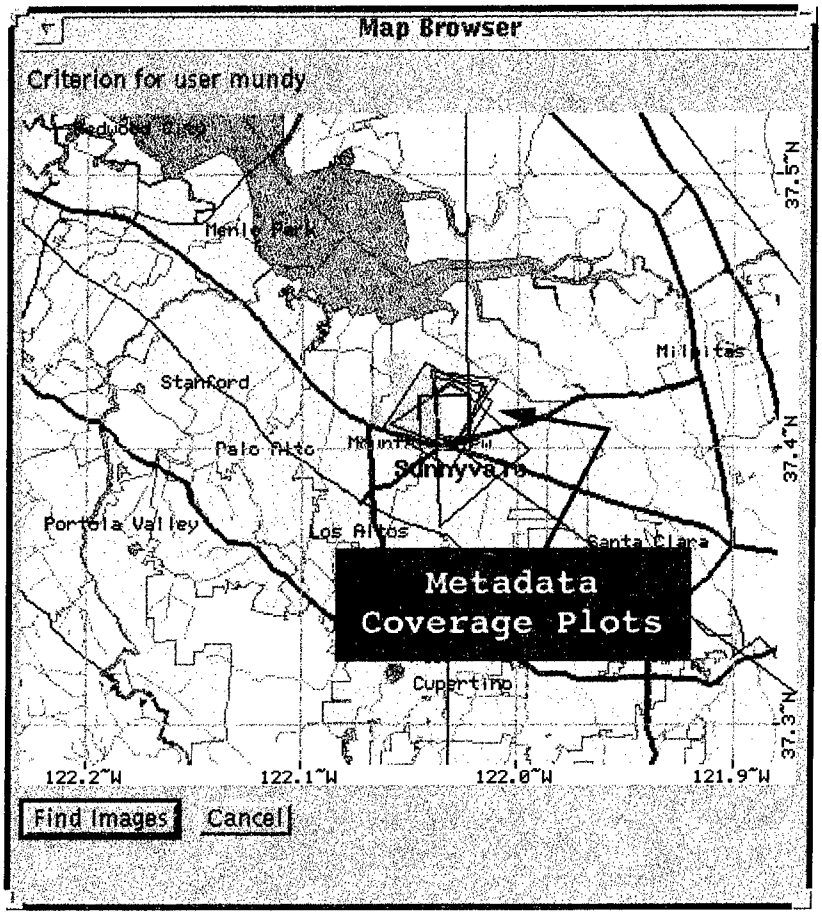


Figure 3. Footprints of aerial images of the Lockheed Martin/ Sunnyvale facility

Figure 4 illustrates how this CBIR formulation process is performed. You will note that, in this image, FOCUS has overlaid a 3D site model of the facility onto a reference image in proper perspective; this assists in CBIR region identification. We call such reference images "content criterion images", as they are used to provide FOCUS with normalcy baselines. An analyst selects a reference region (used to calibrate the underlying image understanding algorithms) and then selects an event region. In this case, the analyst is interested in finding all images in the image archive where Building 158 of the LMMS complex is not visible; this is a typical task performed by intelligence analysts doing 'negation'. A parking lot near Building 158 acts as reference region. The reference and event regions are highlighted in the figure.

In order to perform automated change detection and cueing, FOCUS projects the 3D site model onto each of the images which meet the metadata constraint and "carves out" image chips upon which it runs image understanding algorithms. These algorithms make statistical determinations on the likelihood of change or activity in the region under consideration by comparison with the normalcy baselines; the result of the IU algorithms is a confidence number, between 0 and 1, reporting the estimated change significance of each image. (A ranking of 0 indicates no change from the image initially selected to specify the content criterion; a ranking of 1 indicates an appreciable change from the IA-designated normalcy.)

Figure 4. Using the FOCUS GUI to Specify a Content-Based Criterion for Building Negation

As each image is processed, the "carved out" image chip is stored locally with the confidence number. Once the FOCUS processing is completed on all images meeting the metadata criterion, these "carved out" image chips are assembled and sequenced in order of confidence number to form an "analyst tip sheet". A time-constrained analyst would use such a sheet to assist in workflow prioritization. The IA focuses (hence the name of our effort!) attention on those images meeting the content criterion.

Digital Tip-Off for User payton

- **Structure Method** Building 158 :

  Image Moffett_1970, significance 1

- **Structure Method** Building 158 :

  Image Moffett_1975_Left, significance 1

- **Structure Method** Building 158 :

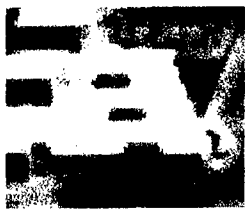
  Image Moffett_1975_Right, significance 1

Figure 5. Top of the Analyst Tip Sheet -- Images where Building 158 is **not** present

Figure 5 shows the 'head' of the tip sheet. The objective of our query was *to retrieve all images where Building 158 was not present*; the first image chips in the list are those images where FOCUS determined that Building 158 was absent (these images were taken in 1970 and 1975 before it was constructed). Figure 6 shows the 'tail' of the tip sheet. Since our query was *to retrieve images where Building 158 was not present*, those images containing the building appear lowest in the list because FOCUS assigned low priority/confidence numbers to them.

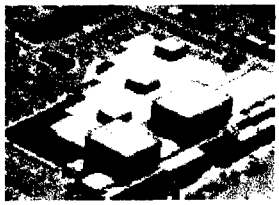
- **Structure Method** Building 158 :

  Image 12_04_90_2x, significance 0.478333



- **Structure Method** Building 158 :

  Image 01_23_89_2x, significance 0.426206



- **Structure Method** Building 158 :

  Image 95-2, significance 0.335149

- **Structure Method** Building 158 :

  Image 12_18_86_2x, significance 0.221138

Figure 6. Bottom of the Analyst Tip Sheet -- Images where Building 158 is present

4 Conclusions/Lessons Learned

We have tested our FOCUS laboratory prototype on a small set of sites. These experiments have confirmed the validity of the FOCUS approach; extensive testing in an operational envionment on larger, diverse sets of images and facilities is required to 'productize' FOCUS. A major objective of the current year's effort is to install and test a FOCUS prototype in an operational image exploitation facility.

Registration of models to images is crucial to FOCUS. Image analysts require that the registration process be automated (thereby reducing analyst time burdens), precise (resulting in accurate model-to-scene association), and rapid (to meet analyst timelines). We are investigating several "registration engines" for their suitability in this phase of the FOCUS pipeline.

It is crucial to extend FOCUS to include a diverse toolkit of image understanding metrics (e.g., edge, texture, and morphological statistics) required to match to high-level image analyst queries. To broaden the applicability of FOCUS to commercial/remote sensing imagery, these algorithms should work in single and multiple band domains.

Acknowledgments

The results reported here would not have been possible without the model-supported exploitation technology base developed during the past decade under RADIUS and associated Government-sponsored image understanding research programs. The authors also wish to acknowledge the internal research and development support for this work provided by the Lockheed Martin Corporation and the Lockheed Martin Advanced Technology Center.

Temporal Analysis of Vehicular Activities from SAR/EO

R. Chellappa P. Burlina
Q. Zheng C. Shekhar L.S. Davis
Center for Automation Research, University of Maryland
College Park, MD 20742-3275 (rama@cfar.umd.edu)

Abstract

We describe a focused research effort whose ultimate goal is the design of an exploitation system capable of temporal analysis of vehicular activities from multisensor data. The purpose of this system is to detect, track and recognize isolated or clustered vehicles, as well as structured or unstructured vehicle groupings and their activities, in heterogeneous sites, for varying rates of observation. Objectives, system components, open research issues and evaluation plans for this project are summarized in this paper. Potential applications are in processing TESAR data as well as in extensions of MSTAR efforts when revisit scenarios are included.

1 Introduction

The proposed research effort involves the design of an image exploitation (IE) system dedicated to vehicular activity monitoring, incorporating context information and temporal reasoning. The purpose of this system is to detect, track and recognize isolated or clustered vehicles, as well as structured or unstructured vehicle groupings and their activities, in both urban sites and open terrain areas.

The proposed system will have two basic modes of operation: a revisit mode and a temporal analysis mode. The revisit mode allows for exploitation of newly acquired synthetic aperture radar (SAR) or electro-optical (EO) images using previously constructed site models. This mode is very close to the types of operational methods envisioned and implemented for the RADIUS project. It extends the RADIUS work by providing additional revisit operations for the exploitation of SAR images, including multi-image SAR-SAR and SAR-site registration, multi-image wide-area site model construction from SAR, and multi-image/multi-resolution detection and recognition of vehicle targets in SAR. This mode has potential applications in processing PM-TESAR data as well as in extensions of MSTAR efforts when revisits are involved. The temporal analysis mode of operation extends the revisit mode by exploiting the latest state of the site along with the site history, site evolution, or object dynamics. This mode includes three basic operations: (a) focusing using temporal inference, (b) generation of activity summaries for targeted functional areas, vehicles, or vehicle formations, and (c) event analysis.

The planned system addresses the necessity for any operational IE system to be capable of tracking movable objects across wide areas. A system with the ability to track single vehicles and to analyze their activities and evolution of vehicle formations over a period of time would be an important asset in both surveillance and tactical applications. Our system allows for exploitation of structured as well as unstructured sites. In structured areas (garrisons, urban environments), such a system would be capable of tracking vehicles and vehicle groupings in parking, storage, and loading areas as well as on roads. In open areas, the system would allow the classification of vehicle groupings, identification of their states of operation, and monitoring of both the groups' overall motions and the evolution of their internal configurations.

The system will integrate the following features: A *context-aided IE* paradigm emphasizing the use of site-model and context information is used for image prioritization, detection, recognition, and monitoring tasks. The system allows for *multisensor IE*, using both SAR (spotlight and stripmap) and EO imagery for detection and recognition. The evolving nature of the observed vehicle formations, and

This project will be supported by the Defense Advanced Research Projects Agency (ARPA Order No. E657) and the U.S. Air Force Wright Laboratory. For further information see http://www.cfar.umd.edu/~akunuri/IE/ie.html.

causality in group and site evolution, require the design of *temporal reasoning* schemes. Temporal analysis will allow prediction, negation, and trend analysis, and will support feedback mechanisms to the detection, classification, and prioritization modules. Temporal analysis will rely on the use of inference systems for event modeling, and will exploit IE results stored in spatiotemporal databases. *Dynamic system modeling* tools will be used for tracking and predicting cluster motion and the evolution of group configurations.

We plan to include the following functional components in our system:

Image prioritization: Context-based delineation and cueing mechanisms will be used to guide the application of monitoring tasks. Focusing will be partially driven by the requests of the Imagery Analyst (IA) according to the current order of mission and by predictions generated by the temporal reasoning modules.

Detection and classification: We will address the detection and classification of isolated target vehicles and of vehicle clusters, both loose and structured, from SAR only and from SAR supported by EO. Detected targets will be clustered into groups using geometrical considerations. We will consider the cases of target groupings constrained to assume periodic formations, target clusters exhibiting regular configurations conforming to standard formations, and unstructured forces organized in loose but repeated cluster subgroups. Different schemes (spectral analysis, rule-based reasoning, or template matching using labeled graph matching) will be used; these schemes will be selected according to context and location as well as by temporal reasoning.

Tracking cluster position and evolution: Region-dependent vehicle dynamics coupled with inference rules will encode the groups' dynamical behaviors. These models will be supplemented with rules describing the allowed macroscopic evolution of a group's configuration (fusion, fission, or reorganization). The road network models, and the presence of possible obstacles or supply and ammunition dumps, will constrain the prediction of hypothesized group positions. These hypotheses will drive focusing and negation mechanisms.

Multisensor image positioning: Algorithms for positioning SAR and EO images to existing site models are necessary for context-based exploitation, delineation, fusion and cueing mechanisms. Novel algorithms incorporating multi-image triangulation, refinement using DEMs, and model-to-image feature matching will be developed.

Wide-area site model construction: The planned system will incorporate tools to construct wide-area site models from multiple SAR observations.

The originality of the proposed research lies in a global approach to the problem of vehicle monitoring and in the use of innovative tools and strategies designed for this purpose. Several key ideas are proposed: (a) A major enhancement over past approaches to the problem lies in *temporal reasoning and multirate temporal exploitation.* The system will be capable of reasoning about and interpreting data acquired at various frequencies. This will be carried out by a combination of inference rules and dynamical systems for describing system evolution. (b) The system will enable *multi-site exploitation*, and will be capable of tracking isolated and clustered vehicles from structured areas to unstructured open areas. (c) The system will use an innovative suite of techniques, including spectral analysis and labeled graph matching, for *group detection and analysis.* (d) The system will expand and integrate the set of tools for revisit operations on SAR imagery, including site construction, target detection and recognition and registration.

This paper is structured as follows: Section 2 outlines our objectives and approach. The various system components and open research problems are reported in Section 3. Section 4 presents our performance evaluation approach.

2 A SAR/EO IE System Guided by Temporal and Contextual Information

The importance of an IE system that could assist the IA in continuous surveillance of vehicular activities is obvious. Such a system should be able to monitor the movement of vehicle units. It should operate across sites of various natures, including garrison areas, where materiel and personnel are stored, as well as on roads and in open areas. It should be able to support the monitoring of large numbers of images collected by UAVs. Processing such large datasets calls for the development of effective focus-of-attention mechanisms, learning and tuning parameter training strategies, as well as adaptive detection and recognition schemes.

Visual imagery alone is not sufficient to effectively address the continuous monitoring of vehicular activities. SAR is the sensor of choice for UAVs; SAR images enable all-weather monitoring, and their acquisition is independent of illumination conditions and range [4]. This is a critical element if one considers that UAVs are likely to offer mostly high altitude observations of the sites under scrutiny. In any case, the joint use of SAR and EO imagery offers an effective set of sensor data for vehicle monitoring purposes from UAV types of platforms. To this end,

we envision a multisensor system where SAR is the primary source of sensor data, with the support of EO data when available.

The system's architecture, depicted in Fig. 1, includes the following functionalities: Images are positioned to an existing site, focusing and delineation mechanisms are applied to determine Regions of Interest (ROIs), followed by detection and classification of targets from SAR or combined SAR-EO observations. The detection/classification results are used for grouping and group analysis. The temporal reasoning modules support several functions ranging from area prioritization to event analysis. Over the last several years, many Image Understanding (IU) algorithms have been developed to support the above exploitation tasks. While they have demonstrated promising capabilities, the application of these algorithms in operational situations has often been limited by their lack of robustness. Three factors will favorably impact the effective use of IU algorithms in surveillance systems; these are the context-aided exploitation paradigm, interactivity, and temporal reasoning.

Systems that track site activities over time can exploit causality in site and event evolution. Forward and backward prediction (backtracking) schemes can be used to support focusing and negation mechanisms. Temporal reasoning can also be used for inferring trends or recognizing events of significant natures. Our system will support temporal analysis and reasoning by using a powerful combination of dynamical system modeling and inference systems—describing the systems' evolution at both coarse and fine temporal scales—coupled with spatio-temporal databases.

The importance of context-based exploitation was demonstrated in the RADIUS project [1]. Site models and context information enable focusing mechanisms. We have shown as part of the RADIUS project that site models, by reducing spatial and spectral search spaces, allow the robust use of spectral analysis methods—known otherwise for their fragility when applied to recognition or texture analysis—for attentional and detection purposes [2]. Exploiting site model information was also very useful in implementing effective vehicle detection and counting algorithms [5]. We show in a companion paper [3] how site models can be used to store valuable training patches exploited by auto-calibration schemes, for automatically tuning IU algorithm parameters. Finally, context-based exploitation emphasizes the development of specialized algorithms whose application is controlled by inference systems guided though interaction with the IA [8].

Interactivity is another key feature of future operational IE systems. Our system will feature several interactive processes (including search, focusing, and target/group classification) in situations where human interaction is truly needed, so as not to impede the processing of large image datasets. Interactivity will be emphasized in bootstrap modes, for the recognition of new vehicles/groups in regions tagged by the IA, and avoided in batch modes.

3 System Components and Research Issues

Overall system description Our IE system is dedicated to vehicular activity monitoring using SAR, possibly supported by EO. This system follows two main operational modes: (a) the revisit mode and (b) the temporal analysis mode.

In the revisit mode, newly acquired images are analyzed using a site-model-based exploitation paradigm, in a fashion reminiscent of the exploitation mode adopted by the RADIUS project. This makes use of up-to-date site models constructed from previously acquired imagery. Our work will extend the RADIUS work by designing revisit functions for the exploitation of SAR images, particularly PM-TESAR data. These functions include: (a) multi-image SAR-SAR and SAR-site registration, (b) multi-image wide-area site model construction from SAR, and (c) multi-image/multi-resolution detection and recognition of vehicle targets in SAR. This mode can also be used for extending MSTAR efforts when revisits are involved.

In contrast with the revisit mode, the temporal analysis operational mode exploits the latest state of the site along with the site history. The evolution of site features relevant for vehicular activities (roads, loading areas, parking lots), or the vehicle dynamics and positions across time, are stored in a dynamic database and retrieved using spatio-temporal queries. Site history is then utilized in specific exploitation scenarios for carrying out three basic functions: (a) focusing using temporal inference, (b) generation of activity summaries for targeted functional areas, vehicles, or vehicle formation, and (c) event analysis. This operational mode will be applied to SAR (PM-TESAR, MSTAR) as well as EO (new Ft. Hood, MB2), possibly with embedded real vehicles staged so as to simulate interesting scenarios.

We plan to investigate the following enabling theoretical and algorithmic tools, system features and innovative techniques in support of the above system functionalities:

Positioning of SAR and EO A system for vehicle monitoring should support fusion and cueing mechanisms among sensors. In an operational system, EO and stripmap SAR images, acquired by HAE

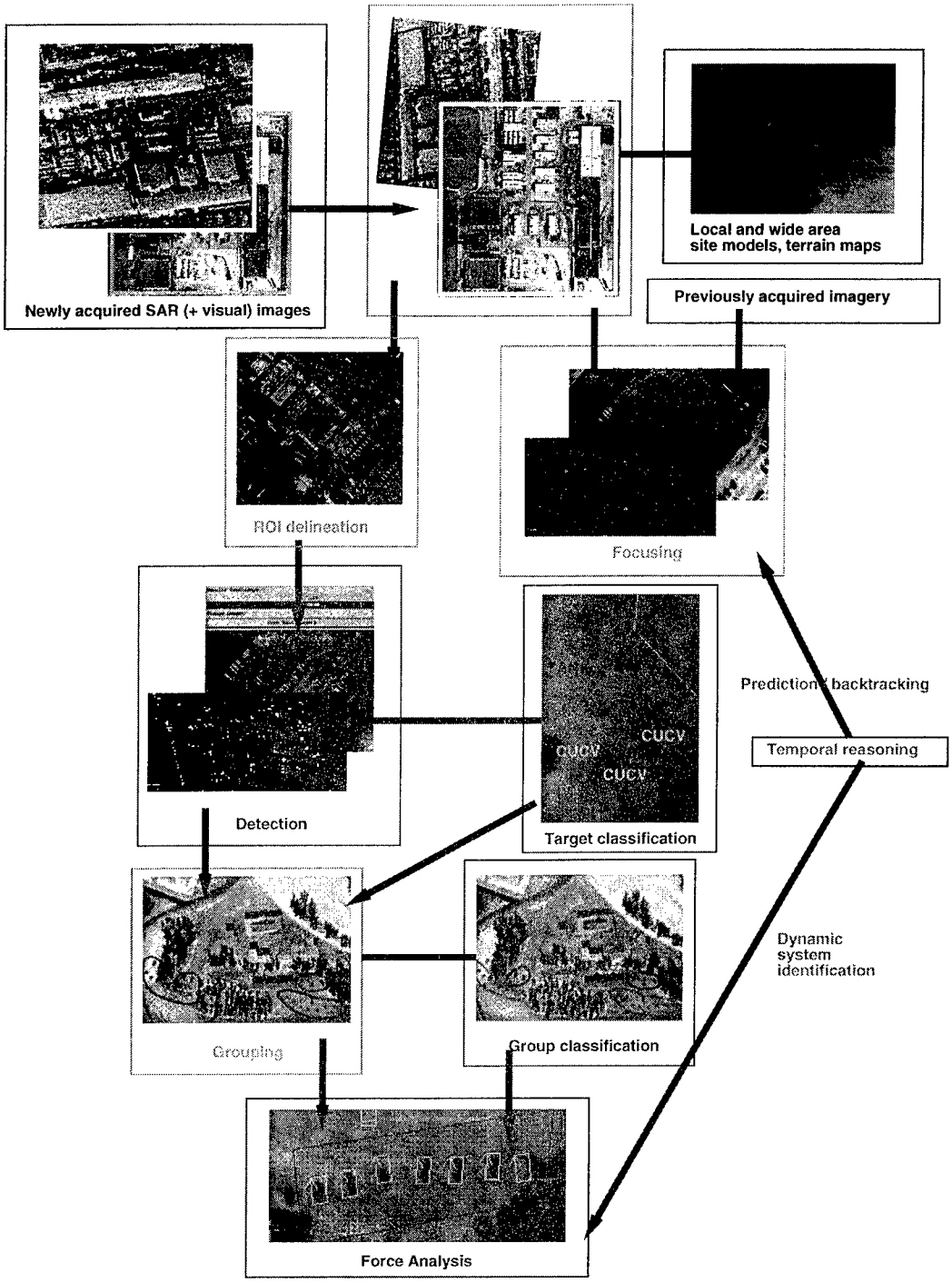
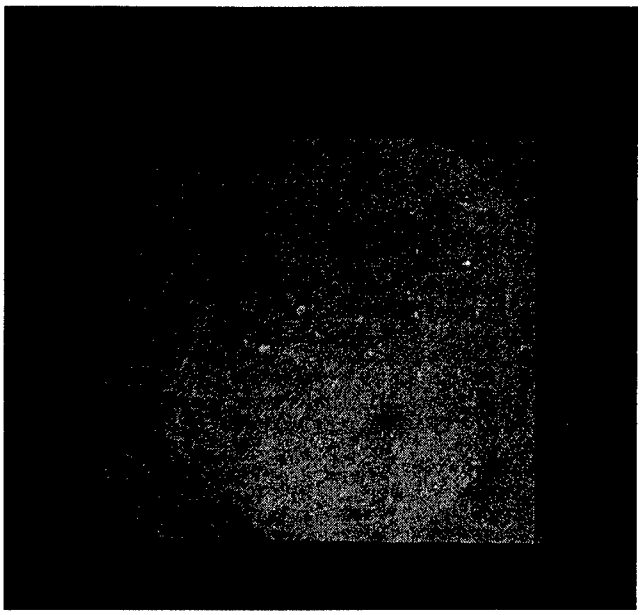


Figure 1: System architecture

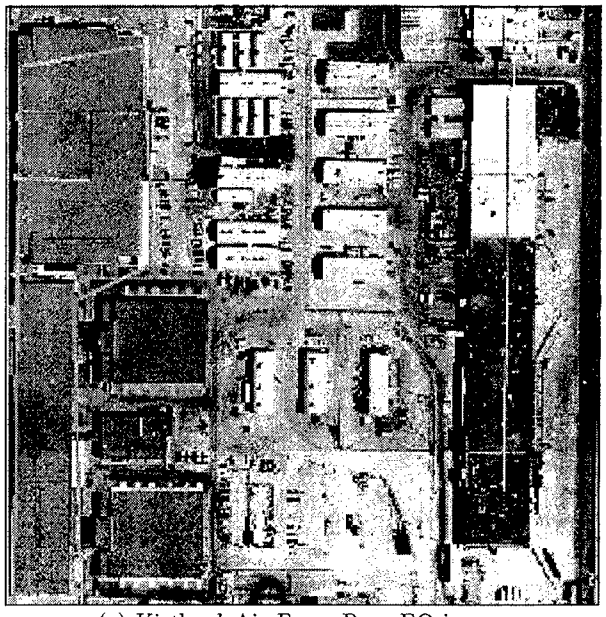
UAVs, can cue the acquisition of higher-resolution EO and spotlight SAR images acquired by aircraft and UAVs. The functions of cueing, fusion of heterogeneous sensor observations, and delineation of ROIs all point to the necessary geometric interaction between SAR and EO images and existing site models, and thus the importance of robust and automated positioning techniques. Note that the registration work within this focused research effort will only support our own needs and will not replicate other efforts in this area.

We depart from image-to-image registration [6; 7] and consider multi-image positioning of SAR and EO imagery to maps (including topographic maps, Digital Feature Maps, or maps derived from SAR), wide-area sites, as well as local-area sites. Fig. 2



(a) Co-positioning of three Stockbridge SAR images

(b) Kirtland Air Force Base SAR image



(c) Kirtland Air Force Base EO image

Figure 2: SAR-SAR and SAR-EO registration

illustrates these tasks. Fig. 2 (a) shows three registered SAR images of the Stockbridge target array set, where an affine warping can be derived from the SAR acquisition parameters (azimuth and slant angles along with range and cross-range resolutions) or image correspondences, if acquisition parameters are not available. Figs. 2 (b) and (c) are respectively EO and SAR images of the Kirtland dataset obtained using a correspondenceless method, where a registered building structure is highlighted on both images.

Positioning with respect to wide-area as well as

local-area maps and 3D sites will be addressed in the context of geopositioning/georeferencing. The critical problem of registration lies in feature correspondence. Robust positioning across photometrically and radiometrically diverse sensors such as SAR and EO needs to be mediated through a common site and necessitates the development of feature- and object-based positioning techniques relying on model-to-image correspondences. Features appropriate for detection are quite different for SAR and EO. We plan to develop sensor-specific tools for detecting cultural as well as man-made features visible in both SAR and EO (such as detectors for building corners or road intersections). Positioning will then be accomplished using model-based matching. Multi-image triangulation will be used to generate additional tie points in unconstrained areas of the image. Simplified slant-plane orthographic projection assumptions will be generalized to include terrain models of increasing complexity, including flat, planar, and full DEM. We will address SAR-specific effects of ground-level projections, variations in elevation, and layover. When coregistered digital elevation maps are available, errors due to layover and shadow will be compensated for. Also, the presence of varying shadows cast in different SAR views will be considered.

Context-aided prioritization The need for applying IU algorithms across large numbers of images requires effective image and region prioritization mechanisms. Image prioritization and locale focusing mechanisms are based on the following mode of operation: ROIs corresponding to the current order of mission are tagged for exploitation by the IA; previously detected groups tagged for continuous monitoring by the IA must be accounted for, to the extent possible, in any subsequently acquired images.

In light of the above considerations, focusing will be driven by the following modules:

- Regions tagged by the IA are delineated for exploitation using site model information.
- Possible locations of vehicles tagged for continuous monitoring are hypothesized by integrating forward the dynamics of the group. This prediction is made on the basis of past observations of tracked single vehicles or formations stored in the spatial database. These hypothesized positions are inferred from dynamical models of the vehicles or groups, encoded in inference rules (for low-frequency observations) or dynamical observer-predictor systems (for high-frequency observations). The hypothesized locations of the monitored groups of vehicles will then provide the needed focus-of-attention mechanism by passing ROIs to our target detection modules.

Note that the group dynamics are learned from past observations and are indexed by the nature of the traversed terrain, the weather encountered, and the presence of hostile objects. Further note that the learned dynamics offer an additional very attractive means of inferring the group's nature or state of operation.

Vehicle detection and classification We will improve our current SAR and EO detection algorithms. Non-Gaussian CFAR target detection results will be filtered by morphological operations. Segmentation will help discard returns from clutter or urban areas (see Fig.3 showing the results of CFAR detection on a Stockbridge target array image). For EO sensors and for the detection of vehicles in open areas, situations involving single or multiple types of vehicles with no dominant orientation and possibly multiple models will be considered. Discrimination/classification of vehicles will be carried out from SAR or joint SAR-EO observations. Target detection will be based on a decision-theoretic approach exploiting an observation space including various target features such as fractal dimension and the Topographic Primal Sketch (TPS). Given the heterogeneous nature of the classification features, classification will be implemented using decision trees. Class hypotheses will be formulated at this level and further disambiguation will occur during the configurational and temporal reasoning phases, where the system dynamics of the observed target will also be used for recognition purposes.

Group detection and recognition Clustering of the vehicle groups will be carried out first. This grouping procedure can rely on simple techniques, or it can include more complex analyses of the grouping geometry. As an example, a simple determination of the convex hull was applied to group the vehicles detected in EO images on roads and open areas, as shown in Figs.4(d)-(f). For increased robustness, geometrical grouping can be applied jointly to co-positioned SAR images; the CFAR targets individually detected on three Stockbridge images (one of which is shown in Fig. 3) are jointly grouped using distance criteria in Figs. 4(a)-(c).

Group recognition will rely on simple group features, such as the size and composition of the group, as well as how target-like the detected vehicles are. Vehicle formations can also be analyzed in terms of how the vehicles need to respond to terrain topography and trafficability. We propose to combine a set of techniques tailored to the context/situation/location in which the vehicle cluster analysis is carried out, and depending on how loose or structured the group-

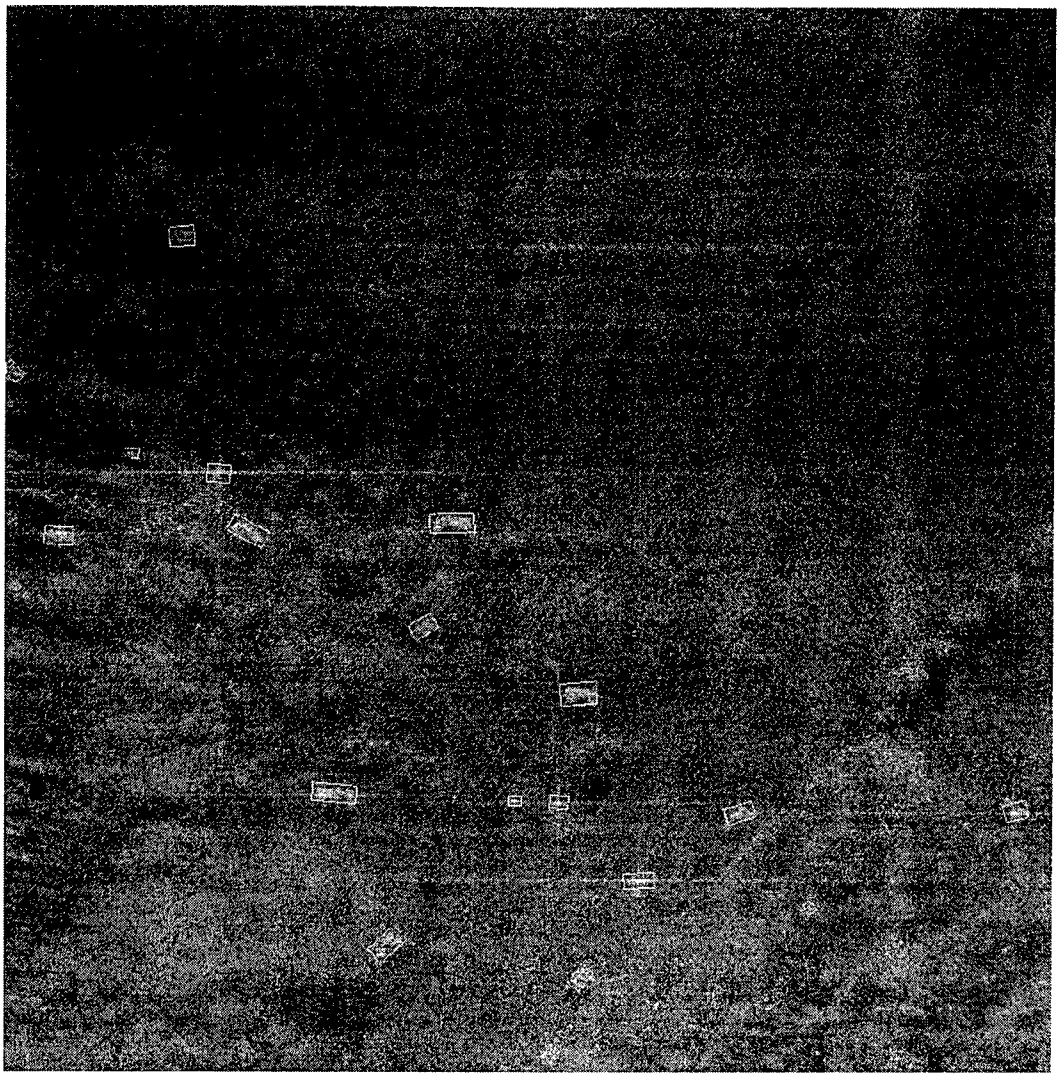
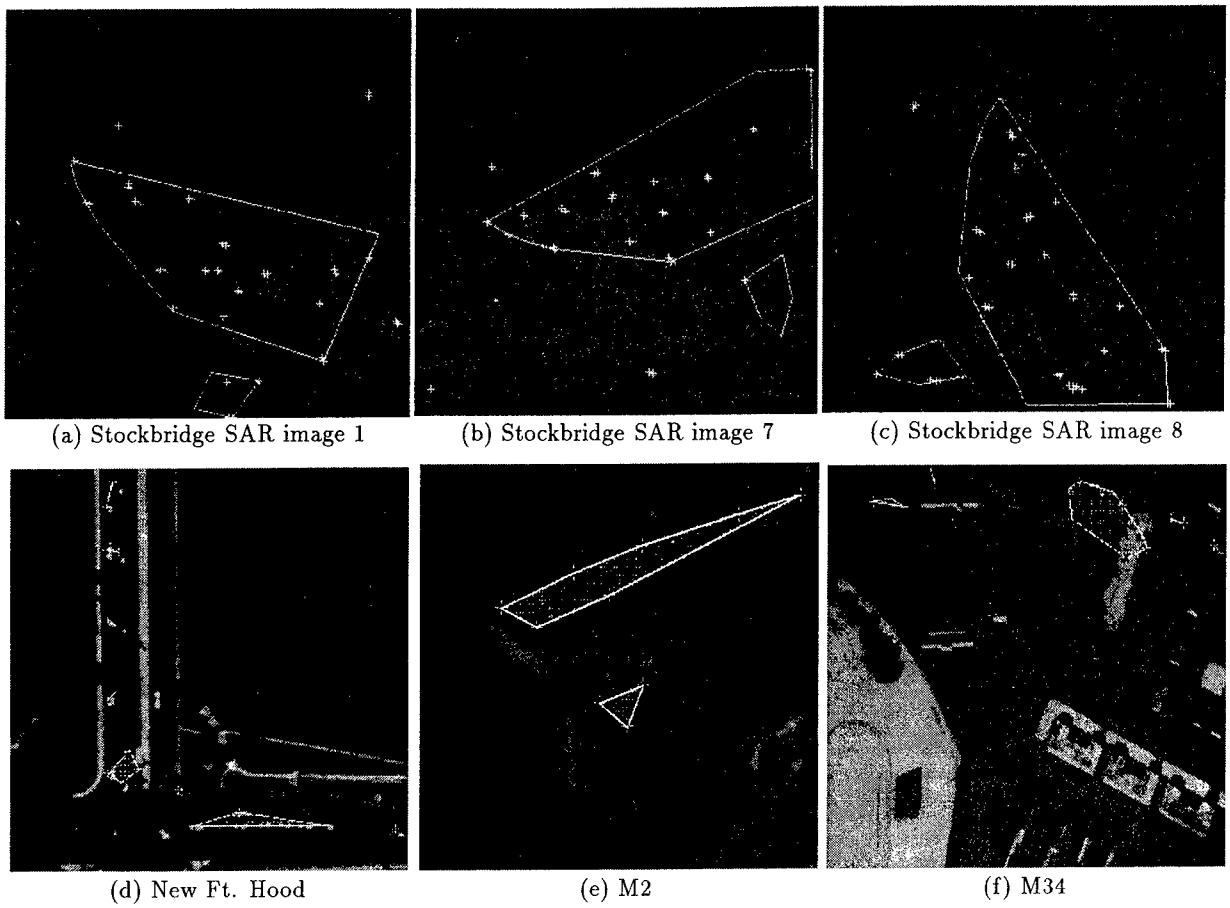


Figure 3: CFAR target detection

ings are. The following cases will be considered: (1) Groups having very regular and periodic formations. This case is relevant for convoy deployments on roads, or stationary vehicles in parking lots or in storage areas. (2) Targets exhibiting regular configurations conforming to standard formations, such as formations of maneuvering units in staging areas. (3) Unstructured groups organized in loose but repeated patterns, such as brigade formations in open areas.

For situation (1), when regular and periodic formations of vehicles are observed, techniques relying on spectral analysis or matched filtering will be used, extending to open areas the mechanism developed in [2] to detect periodic clusters of objects such as convoys on roads or vehicles in parking lots. For situations (2) and (3), a technique relying on spectral analysis for the detection of periodic configurations may not work. In this case, two approaches will be investigated. The first relies on individual target recognition and configurational analysis using inference systems encoding knowledge of deployment configurations. The second uses configurational templates in conjunction with labeled graph matching. Graph matching and configurational analysis will help in determining the operational state and status of the grouping.

Temporal analysis and multiresolution dynamic modeling Temporal analysis serves several purposes and will be addressed at various temporal resolution levels. The first function of temporal analysis is to support one of the focus-of-attention mechanisms previously described. Additional functions of temporal analysis include helping in vehicle/group identification, detecting specific events, carrying out historical trend analysis, change detec-



(a) Stockbridge SAR image 1 (b) Stockbridge SAR image 7 (c) Stockbridge SAR image 8

(d) New Ft. Hood (e) M2 (f) M34

Figure 4: Geometrical grouping of detected vehicles on SAR and EO images

tion, and negation mechanisms.

The ability to infer trends evolving over different time scales is essential. We plan to consider both time-critical changes of a tactical nature, and changes of a possibly slow and progressive nature. This implies that IU algorithms need to be tailored to the selected time scale. For example, if fine change or trend detection is required, selective detection mechanisms for ongoing activities, unstructured groups or unstructured features (such as earth scarring or entrenchment) need to be developed.

The necessity of working at different resolutions also means that specialized tools used for capturing the situation, or more specifically the state evolution of the observed system, need to be employed. One purpose of temporal reasoning mentioned above is to help disambiguate vehicle or group classes. Temporal reasoning can be used to identify single targets or groupings by using the dynamic behavior of these groupings. Temporal rules could encode the manner in which groups evolve at a *coarse temporal resolution*. For example, such rules would encode the evolution of a deployment situation from storage areas to roads, staging areas, open areas, refueling and resupply areas, etc. At an *intermediate temporal resolution*, temporal analysis requires the use of system modeling to capture the dynamic nature of formations and sites. Conditions for the transitions between various states and events for a formation will also be modeled using inference rules. These rules should include combination rules such as fusion, fission of groups, and recombination resulting from incurred losses. These rules will also exploit information derived from context to infer the practicability of routes and paths, and the likely presence of obstacles to ground vehicle movement. For instance, inferences can be made based on the locale position, season, region type, and past significant events (high mountain ranges with snow in the winter, spring flood areas, or locales previously occupied and likely to contain mines, may be designated as impassable). In garrison sites, these rules will encode knowledge of road networks and usual site traffic stored in site models. These will be used along with past observations to predict successive convoy positions for deployment situations.

In addition to vehicle or group identification, the above rules will also help in event analysis, by helping tie together the various group observations to infer events of particular importance. For instance, convoys seen in service areas, then on roads, and then in exercise areas are likely to be engaged in routine exercises. Exercises lasting over periods greater than three days may point to more significant preparations. Similarly, deployments carried out jointly in a large number of garrisons and involving resupply routes are likely to indicate preparations for more specific activities.

Descriptions at *a fine temporal resolution* necessitate the use of discrete event system modeling as well as discrete time system dynamic analysis. These models principally serve the purposes of system identification, as well as prediction and backtracking (backward prediction) mechanisms. These mechanisms will support the focus-of-attention and negation tasks. Focus-of-attention requires being able to hypothesize current locations of previously observed vehicle groupings tagged for monitoring, while negation requires being able to backtrack from the current position of a group to a set of hypothesized points of origin.

The group dynamics depend on the state of affairs of the group (possible handicaps received, availability of fuel, etc.). The group dynamics also very much depend on the nature and topology of the terrain (hills, forest, mountains, roads), which is itself strongly shaped by current and past weather conditions (rain or snow precipitation, ice, etc.), as well as obstacles encountered (river, mines, possible presence of hostile elements). These varying conditions are used in integrating the group dynamics starting from the point where the group was last observed to infer all possible current positions of the group. Similarly, backtracking is implemented by backward integrating the terrain-dependent dynamics of the grouping and taking into account the weather conditions at the time the terrain is crossed to determine the hypothesized set of points of origin of a group of vehicles.

Negation mechanisms are triggered as a result of newly found groups detected in areas tagged by the IA or determined by one of the IU focus-of-attention mechanisms. In negation mechanisms, hypothesized areas of origin are generated for each hour prior to the current detection. Each hypothesized swath is used for querying the spatiotemporal database for image coverage of that swath. If the swath was covered, and analysis of the swath was performed, the database is further queried for any detected group observations on the swath. If no analysis was performed, IU group detection algorithms are applied to these images. This is carried out until candidate groups are produced to which the currently observed group may correspond. If no such groups are produced for the week preceding the current observation, the current group is tagged as unaccounted for.

Another possibility not pursued here, extending this approach, lies in the use of simulation tools for prediction and backtracking purposes. By modeling the evolution of grouping movement and formation as stochastic systems, we will be able to predict the likelihood of possible events, and prepare contingency monitoring plans for the IA.

The above mechanisms will be enabled by a spatiotemporal database holding historical results of past group analyses and supporting trend analyses. This database will support queries made on the basis of spatial proximity, through a DBMS supporting spatial queries.

Interactivity Interactivity yields increased effectiveness in operational scenarios. Interactivity will be concentrated during bootstrap modes, when new vehicles/groups are detected and recognized in regions tagged by the IA. Additional instances where interactivity is critical are as follows: (a) Often IU algorithms rely on many tuning parameters. Most of these can be eliminated using learning techniques and a single parameter representing the detection sensitivity can be saved for tuning by the IA. (b) Control of the algorithms can be driven by a rule-based system interacting with the IA. (c) Focusing and prioritization can be initiated by the IA using a scripting tool according to the current order of mission. The IA should then be able to modify the current set of monitoring tasks in response to detected events. (d) Indeterminacy in vehicle or group classes can be removed by letting the IA choose from a set of most probable candidates.

4 Evaluation and Testing

Continuous experimental evaluation will be carried out so as to ascertain the limits of usability of the IU modules. This is essential for integrating these modules into operational IE systems. Evaluation needs to be carried out at both the module level and the system level to identify the effect of decreased performance of certain modules in subsequent processing stages.

The evaluation of individual module performance will be addressed as follows:

- The detection performance for single targets and target groupings will be quantified using ROC curves. The performance evaluation will be indexed on the number and frequency of target observations.

- The performance of the image positioning algorithm will be evaluated as follows. The matching accuracy can be used to characterize the adequacy of features automatically detected and matched between the image and the site model. The results of multi-image triangulation and tie point generation can be analyzed by using any available ground truth. Lastly, the mutual registrations of image pairs can be evaluated using RMS error.

- The use of temporal reasoning and its advantages can be measured as follows. The gain in performance using temporal reasoning can be measured by assessing (a) the detection performance and (b) the time necessary to detect a previously tracked grouping in a large image by brute force analysis, as compared to using ROIs derived from prediction mechanisms. When using temporal reasoning for identification purposes, the discriminatory nature of observed dynamics will be assessed.

- Grouping algorithms will be evaluated in terms of probability of detection and correct classification.

Imagery for testing our algorithms will include the ADTS, MSTAR, PM-TESAR, Kirtland Air Force Base, new Ft. Hood, and other datasets provided by the government. These datasets will allow us to test detection, classification and grouping capabilities. For more advanced temporal reasoning and group analysis capabilities, we will rely on new datasets in conjunction with semi-synthetic data (real or synthetic targets embedded in real SAR images).

5 Conclusion

We have described an IE framework guided by context information and temporal reasoning. This system is dedicated to the analysis of vehicle groupings observed using multisensor data. The combination of specialized IU algorithms, coupled with robust group modeling and dynamic modeling methods, will lead to an effective system for vehicular activity analysis.

References

[1] P. Burlina, R. Chellappa, C. Lin, and X. Zhang, "Context-Based Exploitation of Aerial Imagery," in *Proc. Workshop on Model-Based Vision* (Boston, MA), June 1995.

[2] P. Burlina, R. Chellappa, and C. L. Lin, "A Spectral Attentional Mechanism Tuned to Object Configurations," *IEEE Trans. on Image Processing*, 1997. To appear.

[3] P. Burlina, V. Parameswaran, and R. Chellappa, "Sensitivity Analysis and Learning Strategies for Context Based Vehicle Detection Algorithms." In these Proceedings.

[4] R. Chellappa, E. Rignot, and E. Zelnio, "Understanding Synthetic Aperture Radar Images," in *Proc. DARPA Image Understanding Workshop*, pp. 229–247, 1992.

[5] R. Chellappa, X. Zhang, P. Burlina, C. Lin, Q. Zheng, L. S. Davis, and A. Rosenfeld, "An Integrated System for Site Model Supported Change Detection," in *Proc. DARPA Image Understanding Workshop*, pp. 275–304, 1996.

[6] R. Chellappa, Q. Zheng, P. Burlina, C. Shekhar, and K. Eom, "On the Positioning of Multisensor Imagery for Exploitation and Target Recognition," *Proceedings of the IEEE*, Vol. 85, pp. 120–138, 1997.

[7] A. Goshtasby, G. C. Stockman, and C. V. Page, "A Region-Based Approach to Digital Image Registration with Subpixel Accuracy," *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 24, pp. 390–399, 1986.

[8] T. Strat, "Integrating IU Algorithms in the RADIUS HUB." Software documentation, 1995.

Image Understanding Research at GE

J.L. Mundy
G.E. Corporate Research and Development
1 Research Circle
Niskayuna, NY 12309

Abstract

Recent progress in image understanding research at GE is described. GE's program in IU is now centered on applications in intelligence image analysis with emphasis on change detection and object recognition. A new effort is now underway to develop descriptions of objects in terms of approximate symmetry to extend the completeness of intelligence event processing.

1 Model-Supported Exploitation(MSE)

We have recently completed work on the Research and Development for Image Understanding Systems (RADIUS) program which is focused on the use of context derived from 3-d site models to enable change detection in intelligence imagery. The work at GE Corporate Research and Development (GE-CRD) has developed or integrated seven change detection algorithms which are now operational in the RADIUS Testbed (RTS) at NIMA.

These algorithms have been used to detect events of intelligence interest on a number of sites and over a limited range of image conditions. The results are sufficiently promising to warrant a more systematic evaluation of the algorithms on a larger image test suite. Additional experience with the performance of these algorithms will be gained during the operation of the Site Monitoring System (SMS), which is a version of RTS being integrated into the Semi-Automatic Image Processing (SAIP) system.

The experience gained from RADIUS has been transitioned to two other Model Supported Exploitation (MSE) applications, PINPOINT and FOCUS. PINPOINT is a system for simulating image formation in weapon target IR sensors, based on accurate 3-d site models and thermal prediction code. Focus is a system for image-based queries on image archives stored in the Image Data Exploitation (IDEX) II soft-copy support system. The results of these queries are used to prioritize imagery to increase effectiveness of exploitation work flow.

1.1 Pinpoint

Our work on geometric and thermal invariants [1], is directed in support of the PINPOINT project. The overall goal of the PINPOINT project is shown in Figure 1. An example simulation produced by the PINPOINT system is shown in Figure 2. A major effort in constructing PINPOINT site models is the assignment of IR material properties. Currently, material type is determined by trial and error using numerical thermal analysis code which solves the heat balance equation. Material choices are deemed correct if thermal analysis yields the observed surface temperatures. These temperature observations are obtained from IR imagery.

In collaboration with Wright Labs [1, 4], we are developing algorithms to classify materials using thermal invariants. Thermal invariants are functions which relate image temperatures

[1]DARPA Contract F33615-94-C-1529

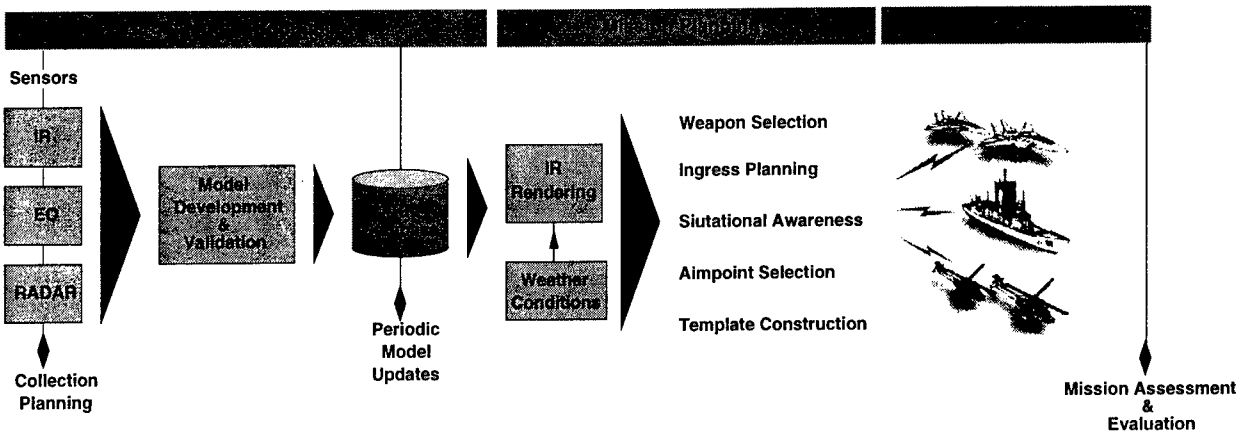


Figure 1: The PINPOINT system concept. 3-d site models are constructed from various image sources. The resulting models are used to predict the appearance of targets in IR weapon sensors. (Figure courtesy of George Gargano, Lockheed Martin, Valley Forge, manager of the PINPOINT project.)

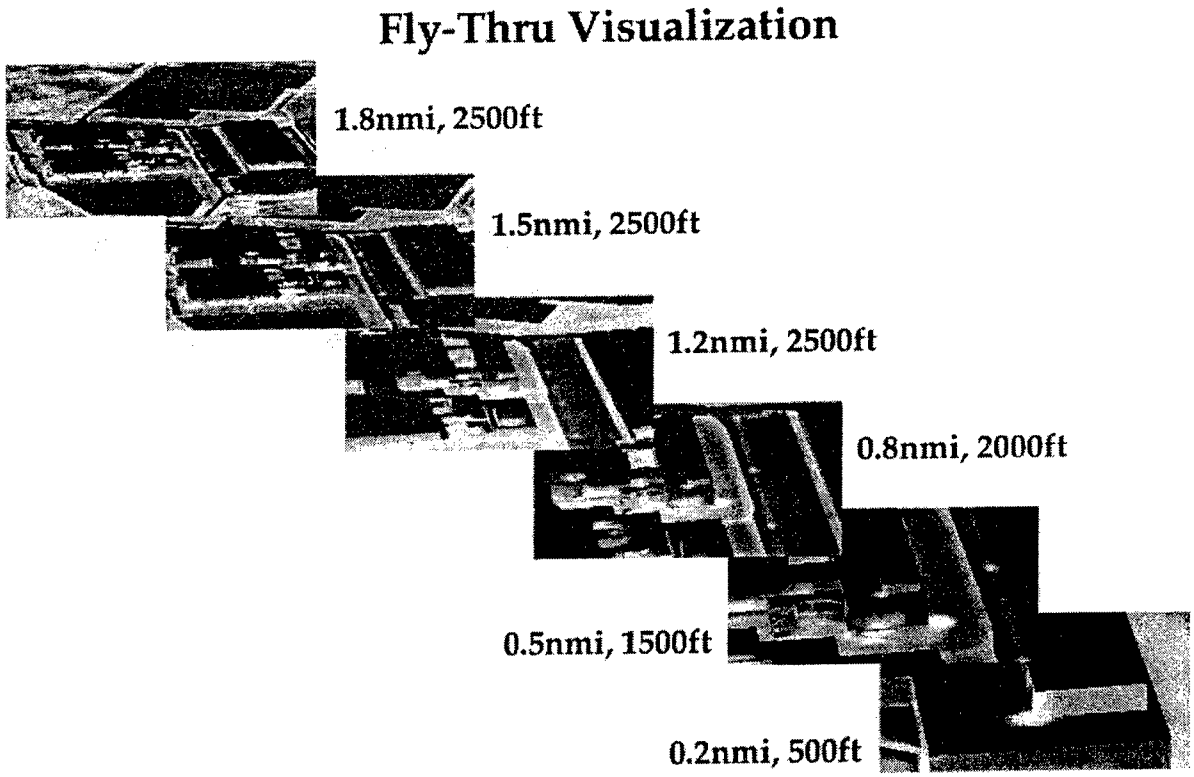


Figure 2: A simulation produced by PINPOINT. A series of rendered images from the 3-d site model depict the views seen by an IR weapon sensor. The IR image values are predicted using thermal modeling, taking into account weather and other conditions that can affect IR appearance. (Figure courtesy of George Gargano, Lockheed Martin, Valley Forge, manager of the PINPOINT project.)

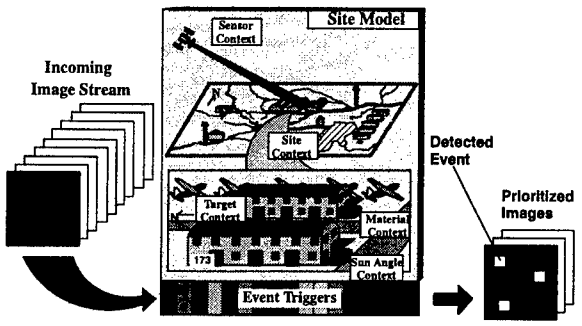


Figure 3:

and some assumed thermal properties to object material properties. Materials can therefore be classified by observing the value of these functions over time sequences of IR imagery. Correct material choices yield the most invariant function sequences. During 1997, we plan to integrate these material classification algorithms with the C++ site modeling system, TargetJr [11], used in PINPOINT. The classification results will be compared with the current manual selection process and validated against thermal modeling results. Some initial trials are reported in these proceedings[5].

1.2 FOCUS

Starting in 1996, an internal research project was initiated to take advantage of the experience gained on the RADIUS project. The goal of the FOCUS system is to prioritize intelligence imagery. The assumption is that significantly more intelligence imagery collection means will become operational by the early part of the 21st century. Increased means of collection include, UAV's commercial satellites and ground-based mobile sensors. Given that an image analyst (IA) is capable of reviewing only a small faction of the available image collection, it is necessary to provide a means for prioritizing the image stream. The concept is illustrated in Figure 3. With this goal in mind, the FOCUS project has developed a web-based query engine which enables the prioritization of imagery based on: image meta-data, such as site location and sensor type; image content using event trigger algorithms developed under RADIUS. The IA can establish event triggers by interacting with a 3-d site model display super-imposed on a reference image of a site. The FOCUS query interface is illustrated in Figure 4. The FOCUS system uses a conventional relational database to initially screen out images based on image meta-data such as site location and sensor type. The images are then prioritized based on the state of event triggers established by selecting site model structures using a site browser implemented in TargetJr. FOCUS acquires either incoming images or archival images using the Output Data Server(ODS) of IDEX II. Work is currently underway to automate image registration and couple to operational meta-data repositories. More details on the FOCUS system are reported elsewhere in these proceedings[2].

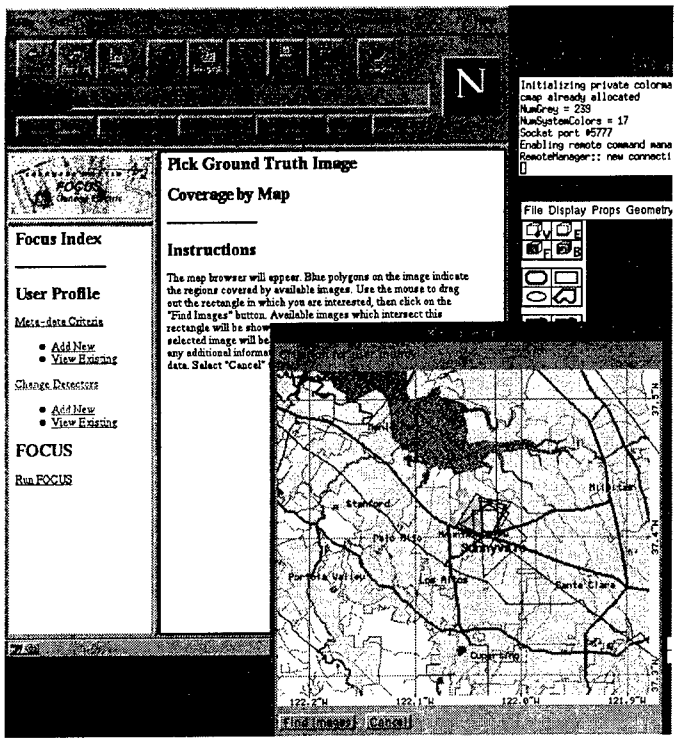
2 Approximate Symmetry

We have just initiated a new project [2] to investigate the use of symmetry as a basis for generic object description for use in intelligence event monitoring. It is common for an IA to wish to restrict a change to a class of objects, such as aircraft. It is also common that IA interest is focused on a specific object, such as a weapon transport vehicle.

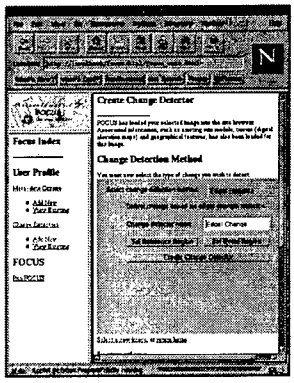
There is currently no way to restrict change in this way without using model-based recognition or other ATR algorithms which require a large number of image observations or a detailed CAD model of the object. Further, the ability of model-based vision techniques to consider classes of objects rather than a specific individual is limited. It is very desirable to acquire the necessary models for detection and recognition of broad object classes with a minimal number of source images and with little effort on the part of the IA.

We are developing an approach to the representation of objects, based on symmetry, which promises to provide such means for modeling and subsequently detecting and recognizing classes of objects[3]. The reason that symmetry is so pervasive is that a symmetrical object

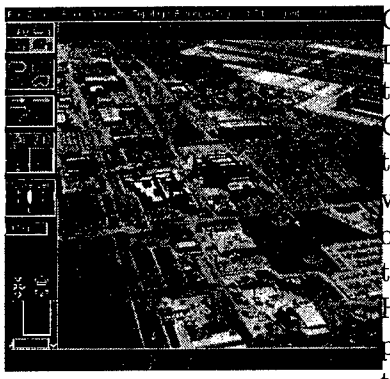
[2]Contract #F33615-97-C-1021.



a)



b)



c)

Figure 4: The interface for FOCUS operation is based on Netscape™. In a) the pre-selection of images is based on image footprints, shown as polygons super-imposed on the site map. In b) a Java applet is used to construct and event-trigger for monitoring an item of intelligence interest. In c) the site-browser tool is shown. The browser is implemented in TargetJr and communicates with Java applets via sockets. A site model for Moffett Field in Sunnyvale, CA is shown.

is both statically and dynamically more stable. In addition, it is economical to repeat structures, as in symmetry, in order to minimize the number of required component designs and manufactured part types. Thus, general object classes can be based on the types of symmetry of any specific object in the class. This class restriction is often sufficient to support event trigger discrimination, because the types of objects, such as vehicles, are limited by the context of the event.

We expect that new descriptions, based on symmetry will enable and IA to describe objects of interest without detailed models and to describe the general characteristics of objects not yet observed. This capability is necessary in discovering new weapon types or structural configurations. An example is shown in Figure 5.

3 The Image Understanding Environment (IUE)

GE CRD is working with Amerinex Applied Imaging to enhance and expand the user interface for the IUE based on a public domain GUI toolkit called FRESCO. Markus Weber of the University of Karlsruhe, Germany, has provided the initial version of the IUE GUI based on FRESCO [12]. This implementation is distributed in the current release of the IUE. Bill Hoffman of CRD is extending this version to provide 3-d rendering and manipulation using the OpenGL library.

Work is also underway to provide an integration of the IUE and TargetJr the C++ environment used in FOCUS and PINPOINT. It is planned that eventually the two systems will be completely merged. A major goal for 1997 is to provide a common user interface across the two environments.

4 Photogrammetry of Pushbroom Cameras

A program called Carmen has been developed for carrying out bundle-adjustment, camera modelling and scene reconstruction from a set of image and scene measurements of general ge-

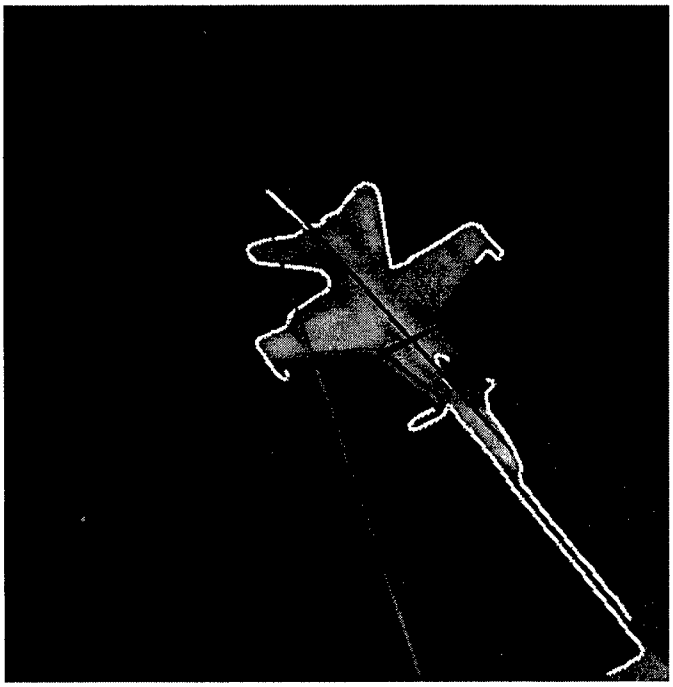


Figure 5: An example of symmetry description. Two feasible transformations corresponding to the bi-lateral symmetry of the aircraft are evaluated. The correct transform produces the black symmetry axis and maps the black curve on the right to the grey curve on the left. The incorrect transform has a much larger error and maps the curve to the shorter grey segment on the tarmac surface. The symmetry axis for this transformation is shown in grey.

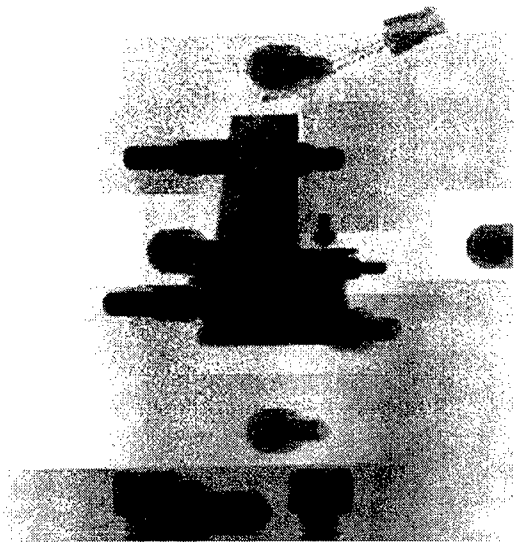


Figure 6: Linear-pushbroom image with spherical marker balls

ometric features. The distinguishing characteristic of this program is that it is able to handle a mixture of arbitrary types of geometric features and camera models. At present, point and line features are supported, as are perspective, pushbroom, panoramic and rational polynomial cameras. Because of the object-oriented nature of the program, it is easily extendible to include very general types of camera, image feature or measurement. Possible geometric features may include plane conics, spheres or more complex geometric models. A Levenberg-Marquardt parameter estimation algorithm is used to optimize the choice of camera and feature parameters to fit the measurements. A structured sparse technique is used to obtain speediest performance on large problems.

An example of the type of image for which Carmen may be used is shown in Fig 6. This is one of a sequence of X-ray images of a turbine blade taken from many angles. The purpose is to reconstruct features of the blade. The application is described in greater details in the papers [10, 9, 6]. The type of sensor used is of the linear-pushbroom type ([7]) which does a central projection in one axial direction and orthographic projection in the other. The characteristics of these X-ray images are similar to satellite pushbroom cameras. We have recently implemented a rational cubic camera using Car-

men. The initial results are described in these proceedings[8].

References

[1] Gauder, M., Velten, V., Westerkamp, L., Mundy, J., and Forsyth, D., "Thermal Invariants for Infrared Target Recognition," Proc. Third Automatic Target Recognizer Systems and Technology Conference, June 1993, GACIAC IIT Research Institute, 10 West 35th Street, Chicago, IL., publishers,

[2] Barrett, E., Peyton, P., and Mundy, J., "FOCUS: A Shared Vision Technology Transfer Project," in these proceedings.

[3] Curwen, R. W. and Mundy, J. L., "Grouping Planar Projective Symmetries," in these proceedings.

[4] Nandhakumar, N. and Velten, V., "Thermophysical Affine Invariants from IR Imagery for Object Recognition," in these proceedings.

[5] Stewart, C., Snell, V., Hamilton, D., Mundy, J., " Thermal Invariants for Material Labeling and Site Monitoring Using Midwave Infrared Imagery: Initial Results," in these proceedings.

[6] Gupta R., Noble, A., Hartley, R., Mundy, J., and Schmitz, A. "Camera calibration for 2.5-D X-ray metrology." In *Proc. International Conference on Image Processing ICIP-95, Volume III*, pages 37 – 40, 1995.

[7] Hartley, R. and Gupta, R., "Linear pushbroom cameras." In *Computer Vision - ECCV '94, Volume I, LNCS-Series Vol. 800, Springer-Verlag*, pages 555–566, May 1994.

[8] Hartley, R. and Saxena, T., "The Cubic Rational Polynomial Camera Model," in these proceedings.

[9] Noble, A., Gupta, R., Mundy, J., Schmitz, A., Hartley, R. and Hoffman, W., "CAD-based inspection using X-ray stereo." In *Proc. IEEE Robotics and Automation Conference*, 1995.

[10] Noble, A., Hartley,R., Mundy,J. and Farley, J. "X-ray metrology for quality assurance." In *Proc. IEEE Robotics and Automation Conference*, 1994.

[11] Snell, V., Mundy, J., et al. "TargetJr Home Page" http://www.balltown.cma.com/TargetJr/

[12] Markus Weber: *Entwurf und Realisierung einer objektorientierten graphischen Benutzungsschnittstelle mit Möglichkeit zur Laufzeitanalyse von Algorithmen für den Bildauswertungs-Systemansatz IUE.* Diplomarbeit, Institut für Algorithmen und Kognitive Systeme, Fakultät für Informatik der Universität Karlsruhe (TH), Februar 1997.

Continuous Terrain Modeling from Image Sequences with Applications to Change Detection

Yvan G. Leclerc*
Artificial Intelligence Center, SRI International
333 Ravenswood Ave., Menlo Park, CA 94025
E-MAIL: leclerc@ai.sri.com
HOMEPAGE: http://www.ai.sri.com/~leclerc/
PROJECT HOMEPAGE: http://www.ai.sri.com/~meshes/

Abstract

The objective of this project is to develop methods to incrementally model and detect changes in the shape or surface material properties of terrain. The model will be derived from range data (such as interferometric synthetic aperture radar (IFSAR) elevation data) and electro-optical (EO) and infrared (IR) imagery. The imagery can be either still images or video data, such as that obtained from the Predator UAVs. The modeling and change detection algorithms will be based on an extension of our object-centered "deformable mesh" approach that incorporates surface material properties and appropriate error estimates.

1 Introduction

The deployment of various monitoring platforms, such as the Predator UAV, will generate large quantities of SAR/IFSAR and EO/IR data of great value to Battlefield Awareness if it can be interpreted quickly and affordably.

We propose to develop and demonstrate a system that will automatically generate and refine a 3-D model of the terrain's shape and surface properties from IFSAR, EO, and IR data, and detect changes in both elevation (due perhaps to bomb damage, movement of large machinery, deforestation, and so on) and surface properties (due perhaps to change in ground cover, pouring asphalt over a dirt road, building an air strip, and so on). Such changes can then be noted on the model for review and action. We thus expect to be able to dramatically reduce the amount of analyst time necessary to take advantage of this type of data.

2 Overview of Our Approach

We propose to use our object-centered "deformable mesh" representation to combine radar and EO/IR imagery taken at different times of day and from different points of view into a unified 3-D model of the shape and surface properties of the terrain [Fua and Leclerc, 1996, Fua and Leclerc, 1995, Fua and Leclerc, 1994a, Fua and Leclerc, 1994b, Fua and Leclerc, 1993].

In this approach, the terrain is represented by a 3-D surface model composed of interconnected triangles called a "mesh." Each triangle, or facet, of the mesh represents an estimate of the position, shape, orientation, and surface material properties (e.g., color, radar reflectance) of the terrain's surface over a small triangular area. The mesh is used not only as the representation of the terrain, but is also an integral

*This work was sponsored in part by the Defense Advanced Research Projects Agency under contract F33615-97-C-1023 monitored by Wright Laboratory. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the United States Government, or SRI International. Many thanks to Pascal Fua for the many years of collaboration we have had in developing the approach described here.

part of the computational framework. Figure 1 illustrates the mesh representation and shows a mesh constructed from a stereo pair of EO images.

The method involves the following components.

Model Creation First, maps, IFSAR elevation data, EO/IR imagery, and other sources of information (e.g., terrain type, building models) will be combined to create a complete 3-D model of the shape and surface properties of the geographical area covered by the data. The surface properties of the terrain will be estimated from the imagery based on the known position of the sun, the shape of the terrain (taking shadows and occlusions into account), camera and radar parameters, cloud cover, and other relevant information. Known "deficiencies" of the sensors (such as occluded areas in EO/IR imagery or "front-porch" artifacts in IFSAR data) will be used to rigorously derive error tolerances and covariances for every element of the model. (In the later years of this effort, we expect that SAR data will be directly integrated into the model.)

Change Detection Second, new imagery and new IFSAR elevation data will be compared against the terrain model to detect changes in the terrain. It is the integrated 3-D nature of our representation and processing methodology that will allow us to detect changes in both the shape and surface material properties of the terrain, as follows.

Changes in the terrain's shape will be detected by comparing 3-D shape and material properties derived from incoming data against the model. This can be done directly for incoming IFSAR range data. For incoming EO/IR imagery, our mesh-based terrain modeling algorithm will be used to register and derive a new 3-D model from the imagery. This derived model will then be compared against the current model, using the error tolerances mentioned above to detect areas of significant change.

Model Refinement Third, new imagery and elevation data will be used to continuously refine the terrain model wherever the new imagery is consistent with the model (i.e., when the elevation data and surface properties derived from the new imagery are within the automatically derived error tolerance of the model). This will allow the model to become increasingly accurate and reliable over time. As the model becomes more accurate, it will support more sensitive change detection.

Model Extension Finally, incremental extensions of the model to new areas will be made wherever IFSAR range data or overlapping images cover a portion of the terrain that has not yet been modeled.

In the following sections we describe our approach and the proposed processing steps in more detail.

3 Mesh-Based Optimization

In mesh-based optimization, information from elevation data and imagery is integrated using a unified optimization framework in which a global objective function is minimized. Each source of information is modeled using a distinct objective function that relates the information to the shape and surface material properties of the surface mesh. A weighted sum of the objective functions is minimized to arrive at a model that incorporates all the information.

We propose to extend our current mesh-based approach in two ways so that it can be used for long-term model building and change detection based on IFSAR range data and sequential imagery. First, we propose to include rigorously derived error tolerances and covariance matrices that specify the range of positions/surface properties of each facet. Second, we propose to modify the optimization procedure so that new data will be processed in sequence as it arrives, rather than waiting for all the data to arrive before processing it.

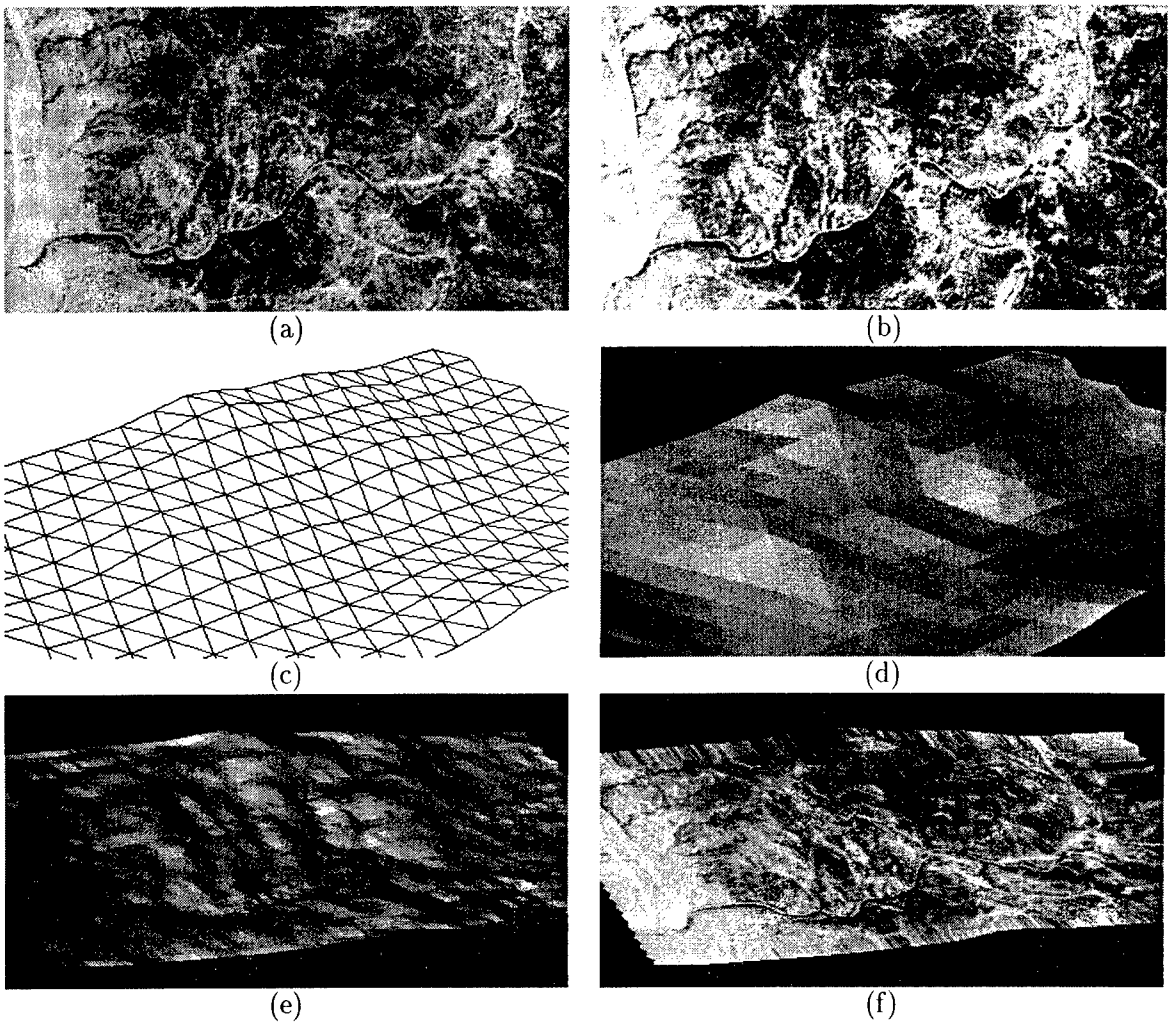


Figure 1: Terrain modeling at the Ft. Irwin, CA, National Training Center (NTC) using deformable meshes. (a,b) A stereo pair of a hilly site. (c) A coarse hexagonally triangulated mesh, shown as a wireframe. (d) A shaded view of the same mesh. (e) The mesh after subdivision and optimization, shown as a shaded surface. (f) The optimized mesh shown with one of the images overlaid on the surface.

3.1 Measures of Uncertainty

We propose to include a rigorously derived error tolerance for every element of the mesh that specifies the range of positions and surface material properties that is consistent with the quantity and quality of data processed to date. For example, areas that are covered by many views would generally have a smaller error tolerance than areas that have been viewed only a few times. Another example is that the elevation in an IFSAR shadow area is not well defined: the minimum elevation is unconstrained, but the maximum elevation can be computed from the look-angle. These constraints form the error tolerances for the shadowed area. Error tolerances can also be determined from other information sources that can be used to augment the model-building process, such as maps or annotations associated with the IFSAR range data. The rigorous derivation and use of error tolerances will be a significant component of our research effort.

In addition to the error tolerance, each element of the mesh has an associated covariance and information matrix. The covariance matrix represents the degree of uncertainty in the element's

state. It is closely related to the error tolerance, and is directly related to the shape of the potential surface (defined by our objective function) in the neighborhood of the current estimate. The information matrix is the sum of the inverse covariances of the data used to update the element. As new data arrives, its information matrix is added to the information matrix of the associated model element.

3.2 Model Initialization

An initial terrain model must be created before refinement and change detection can take place. If a reasonable estimate of the shape of the terrain is provided (from IFSAR range data or maps, for example), then the initial model creation can be done automatically. Otherwise, manual intervention will be required. For more information on how manually derived information is used to initialize a mesh, see [Fua and Leclerc, 1995].

3.3 Data Processing

Once an initial model is created, incoming IFSAR range data and imagery will be used in three ways: to detect and correct for errors in the model, to detect changes in the area, and to update the terrain model to make it more accurate.

Incoming imagery will be processed as follows. Incoming imagery (both elevation and EO) is used to create an updated terrain model. This updated model is compared against the current model and corresponding error tolerances. Areas that fall outside of the error tolerances are candidates for model correction or denote changes to the scene. Areas that are inside the error tolerance, on the other hand, are used to refine the model with the optimization process.

For IFSAR range data, the comparison is relatively straightforward. Every point of the updated elevation data is directly compared against the current surface mesh. If it is within the error tolerance, then that part of the data can be used to refine the surface mesh by using a standard Kalman filtering approach. In that approach, the coordinates of the model element are replaced by a weighted average of the model point and the new elevation data, where the weights are the information matrices described earlier.

Isolated points or small areas that fall outside of the error tolerances are likely to be sensor errors that will be ignored. Large areas that fall outside of the error tolerances indicate that either the terrain has changed, or that the mesh is in error. One way to distinguish between these two cases is to re-optimize the mesh using previous imagery. If the mesh changes (that is, the previous imagery is incompatible with the new model, and hence is incompatible with the new imagery), then this indicates that the terrain has changed.

Comparison of EO imagery is more complex. The scenario we envision for incoming imagery is that the images will be processed as sequences in which adjacent frames are taken relatively close together (such as would be the case for UAV video data of a continuous fly-over). A large gap (in time) between adjacent frames will be treated as the beginning of a new sequence, and the remaining frames will be processed as a separate sequence.

Sequences of EO images, as defined above, can be used to refine or detect changes in either the shape or the surface material properties of the terrain. In all cases, the basic idea is to use the sequence to estimate an updated terrain model and then compare this updated terrain model against the current model.

The updated terrain model will be computed by starting with the current model as the initial estimate of our optimization procedure. Each new frame of the incoming imagery will then be used to refine this updated model, using a sequential version of our optimization procedure. Eventually, the covariances of the updated model will become small enough to allow a meaningful comparison against the current model. Since the updated model will contain both the shape and surface reflectance properties, it will be possible to detect changes in both of these aspects of the terrain.

When the sequence is complete, the updated and current terrain models will then be merged (and the covariances appropriately adjusted) wherever the differences are within the error tolerances. Note that, over time, areas viewed multiple times will tend to have lower covariances. Consequently, the change detection will become more sensitive over time.

4 Advantages of the Mesh Approach

Deformable meshes have a number of distinct advantages over traditional image-based stereo and change detection techniques.

- Occlusions in arbitrary views are naturally accommodated because meshes are a full 3-D representation of a terrain. Traditional stereo techniques, on the other hand, require that occlusions be detected explicitly in the images, which is an open research problem.

- Information from many modalities can be naturally integrated within the unified optimization framework. This approach produces a model in which all the information is used together. This is significantly more accurate and robust than traditional processing where, for example, independent depth maps are recovered from stereo pairs and the maps are then "averaged" together in some fashion.

- Constraints from various external sources, such as maps, can be incorporated by creating an appropriate objective function or by constraining the optimization process in the relevant manner.

- The expected accuracy and known artifacts of various sensors can be incorporated into the modeling process by appropriately weighting components of the objective functions. For example, "shadowed" areas in IFSAR data would be weighted very lightly, while data from flatter areas would be weighted more heavily. In addition, error tolerances and covariance matrices specifying the range of positions for the surface elements can be derived from the expected accuracy of the sensors.

- Change detection using new images can be accomplished even when viewpoints and time of day have changed because the terrain model is fully 3-D and includes surface properties. Change detection based on a simple comparison of images (such as current mosaicking techniques) cannot be used for this purpose.

5 Evaluation Plan

We will provide metrics to evaluate the accuracy, robustness, and completeness of the terrain models we produce, as well as the robustness and accuracy of the change detection.

- **Accuracy of the model.** The accuracy of the terrain model will be measured against a number of standards: points on the terrain with known positions (as obtained via Global Positioning System (GPS) sensors on the ground), selected points in images for which the best manual photogrammetry has been applied, carefully monitored automatic stereo analysis systems, and IFSAR elevation data in areas for which IFSAR data had not been supplied to the system.

- **Robustness of the model.** Robustness will be measured in terms of the area of the recovered terrain in which the system made clear mistakes (again as compared to human-recovered terrain models).

- **Completeness of the model.** Completeness will be measured in terms of the area of the recovered terrain for which the system had at least two views but that was not modeled.

- **Accuracy and robustness of the change detection.** The accuracy and robustness of the change detection will be measured by the number of missed changes and the number of false positives generated by the algorithm.

6 Summary

In summary, we propose to develop methods to incrementally model and detect changes in the shape or surface material properties of terrain. This will be done by extending our current mesh-based optimization approach in two ways, so that it can be used for long-term model building and change detection based on IFSAR range data and sequential imagery. First, we propose to include rigorously derived error tolerances and covariance matrices that specify the range of positions/surface properties of each facet. Second, we propose to modify the optimization procedure so that new data will be processed in sequence as it arrives, rather than waiting for all the data to arrive before processing it.

We have proposed a number of methods for robustly detecting changes in 3–D meshes using our approach. These proposed methods are still in the preliminary stages of development, and we will certainly be considering other recent work in change detection to see if some of the techniques can be applied to 3–D meshes [Huertas and Nevatia, 1996, Bejanin *et al.*, 1994, Chellappa *et al.*, 1994].

References

[Bejanin *et al.*, 1994] M. Bejanin, A. Huertas, G. Medioni, and R. Nevatia. Model validation for change detection. In *Proceedings of the DARPA Image Understanding Workshop*, pages 287–294, Monterey, CA, November 1994.

[Chellappa *et al.*, 1994] R. Chellappa, Q. Zheng, L. S. Davis, C. L. Lin, X. Zhang, C. Rodriguez, A. Rosenfeld, and T. Moore. Site-model-based monitoring of aerial images. In *Proceedings of the DARPA Image Understanding Workshop*, pages 295–318, Monterey, CA, November 1994.

[Fua and Leclerc, 1993] P. Fua and Y. G. Leclerc. Combining Stereo, Shading and Geometric Constraints for Surface Reconstruction from Multiple Views. In *SPIE Workshop on Geometric Methods in Computer Vision*, pages 113–123, San Diego, CA, July 1993.

[Fua and Leclerc, 1994a] P. Fua and Y. G. Leclerc. Registration Without Correspondences. In *Conference on Computer Vision and Pattern Recognition*, pages 121–128, Seattle, WA, June 1994.

[Fua and Leclerc, 1994b] P. Fua and Y. G. Leclerc. Using 3–Dimensional Meshes To Combine Image-Based and Geometry-Based Constraints. In *European Conference on Computer Vision*, pages 281–291, Stockholm, Sweden, May 1994.

[Fua and Leclerc, 1995] P. Fua and Y. G. Leclerc. Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading. *International Journal of Computer Vision*, 16:35–56, September 1995.

[Fua and Leclerc, 1996] P. Fua and Y. G. Leclerc. Taking Advantage of Image-Based and Geometry-Based Constraints to Recover 3–D Surfaces. *Computer Vision and Image Understanding*, 64(1):111–127, July 1996. Also available as Tech Note 536, Artificial Intelligence Center, SRI International.

[Huertas and Nevatia, 1996] A. Huertas and R. Nevatia. Detecting changes in aerial views of man-made structures. In *Proceedings of the DARPA Image Understanding Workshop*, pages 381–388, Palm Springs, CA, February 1996.

USC RADIUS Related Research: An Overview

R. Nevatia and A. Huertas*
Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, California 90087-0273
e-mail: nevatia@usc.edu

Abstract

This paper provides an overview of recent work on USC's RADIUS related research projects. Our major project is in change detection and site model updating. We have made significant progress in validating building models and in detecting changes such as changes in dimensions of a building, missing buildings, and detection of new buildings. We also describe new results in our monocular and multi-view building detection and description systems. We also briefly describe methods for minimal user interaction with these systems to improve the quality of the results.

1 Introduction

A key goal of the RADIUS project is to provide tools to assist an image analyst in analysis of large amounts of imagery. It has been established that use of 3-D *site models* is an essential component of this process [Gerson and Wood 1994]. The site models are useful in a variety of ways: in projecting the expected structures from the current view point, in assisting with the task of change detection and in cueing an analyst to the appropriate parts of an image. While utility of the 3-D site models is generally accepted, their initial construction from images and subsequent updating of them is a considerable task in itself.

Our major RADIUS related project is on change detection and site model updating with particular focus on stationary structures. Some of these techniques are also applicable to automatic site modeling and some of our change detection techniques may also apply to detection of larger mobile objects such as aircraft [Marouani et. al. 1995]. We have also incorporated two new processing modes to the interactive modeling system [Heuel and Nevatia 1995] that works in conjunction with our automatic system to minimize the need for tedious interaction. An overview of these projects is given below. Several other papers provide more details ([Huertas et. al. 1995, Lin et. al. 1994, Lin and Nevatia 1996, Noronha and Nevatia 1997]).

2 Change Detection

The task of change detection is to find significant changes that have taken place at a site since the time of last analysis. Note that the interest is in changes in the *site* and not in the *image*. Images can change for several other reasons such as change in viewpoint, illumination and seasons which may not be significant for analysis. Thus, we should compare a new image (or images) with the information contained in what is called a *site folder* which may consist of a site model, results of previous analysis, previous images and any other available collateral information.

Our approach is illustrated schematically in Figure 1. It consists of the following steps:

Site Model to Image Registration: In this step, the initial camera model is refined so that the site model is brought into close correspondence with the observed image. Our method consists of matching image and model line segments for this task.

* This research was supported in part by Contract No. 76-93-C-0014 from the Defense Advanced Research Projects Agency (DARPA) an monitored by the Topographic Engineering Research Center of the U.S. Army, and by DARPA Contract No. F49620-95-1-0457 monitored by the US Air Force Office of Scientific Research

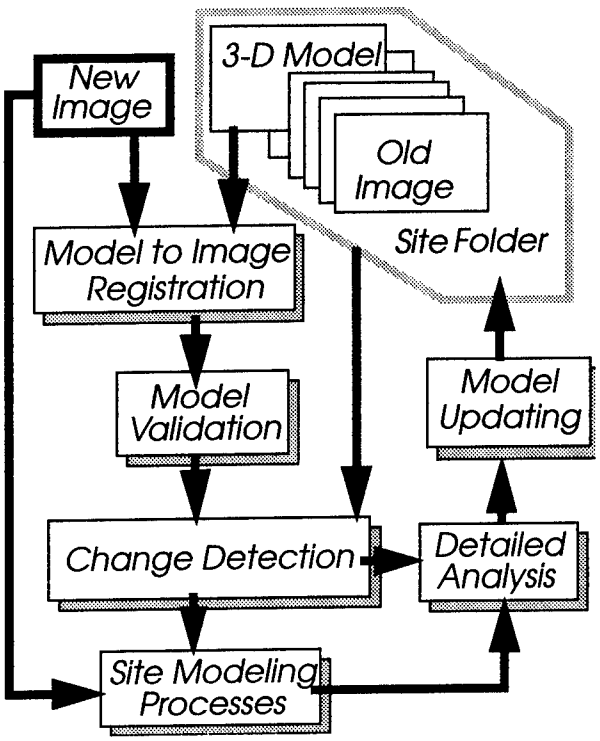


Figure 1 Flowchart of the Change Detection System

Model Validation: In this step, we verify the presence of the model objects in the image based on the correspondence between image features and model features. Locations and objects for which good matches can not be found indicate possible sites for changes.

Change Detection: In this step, we analyze in more detail the possible change sites indicated in the previous step and determine if the missing correspondences can be explained in other ways. The possible detected changes can then be indicated to a human analyst or further analyzed by an automatic system.

Site Model Updating: In this step, the changes are modeled and incorporated in the new site model.

Detailed Analysis: More detailed analysis of the changes can be performed at this stage. For example, to determine the time when the change may have first occurred (*negation*) or to find a sequence of events that leads to the observed change.

We have currently implemented the steps of registration, model validation, a partial capability to detect and describe certain kinds of changes such as buildings which are missing or whose dimensions may have changed. An example of this system is shown below.

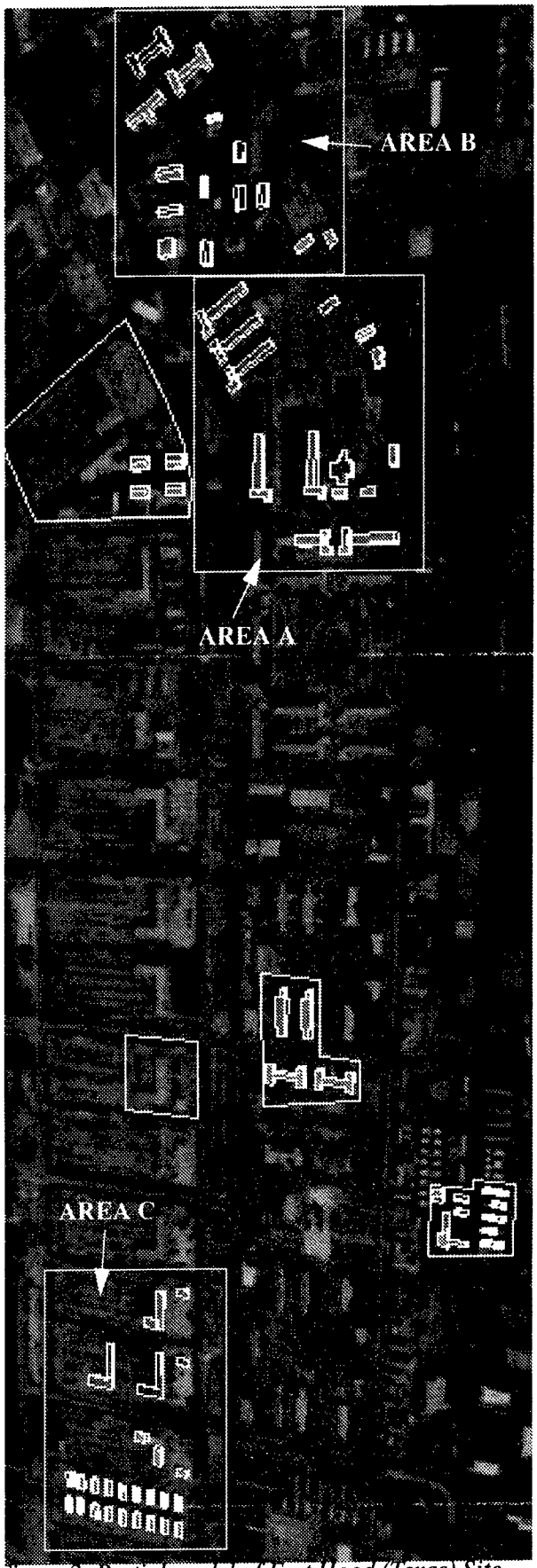


Figure 2 Partial model of Fort Hood (Texas) Site

Figure 2 shows a portion of an image of the RADIUS Fort Hood, Texas, site, with a partial model overlaid. Figure 3, Figure 4 and Figure 5 show the change detection result for three portions of this image (AREAS A, B and C). The confidence level, high (H), Medium (M) and low (L) denotes validation confidence, that is, a reflection on the amount of underlaying image support for the model buildings. Buildings may however have changed in their dimensions (denoted by a circle on the roof), when evidence of change in dimensions can be explicitly found, or when the image support is small. Low confidence levels is, in general, a good indication of major changes, or, in some cases that the building is no longer present, or that the model is inaccurate.

The results for the 79 structures in the model shown in Figure 2 are summarized in Table 1. Of the 79

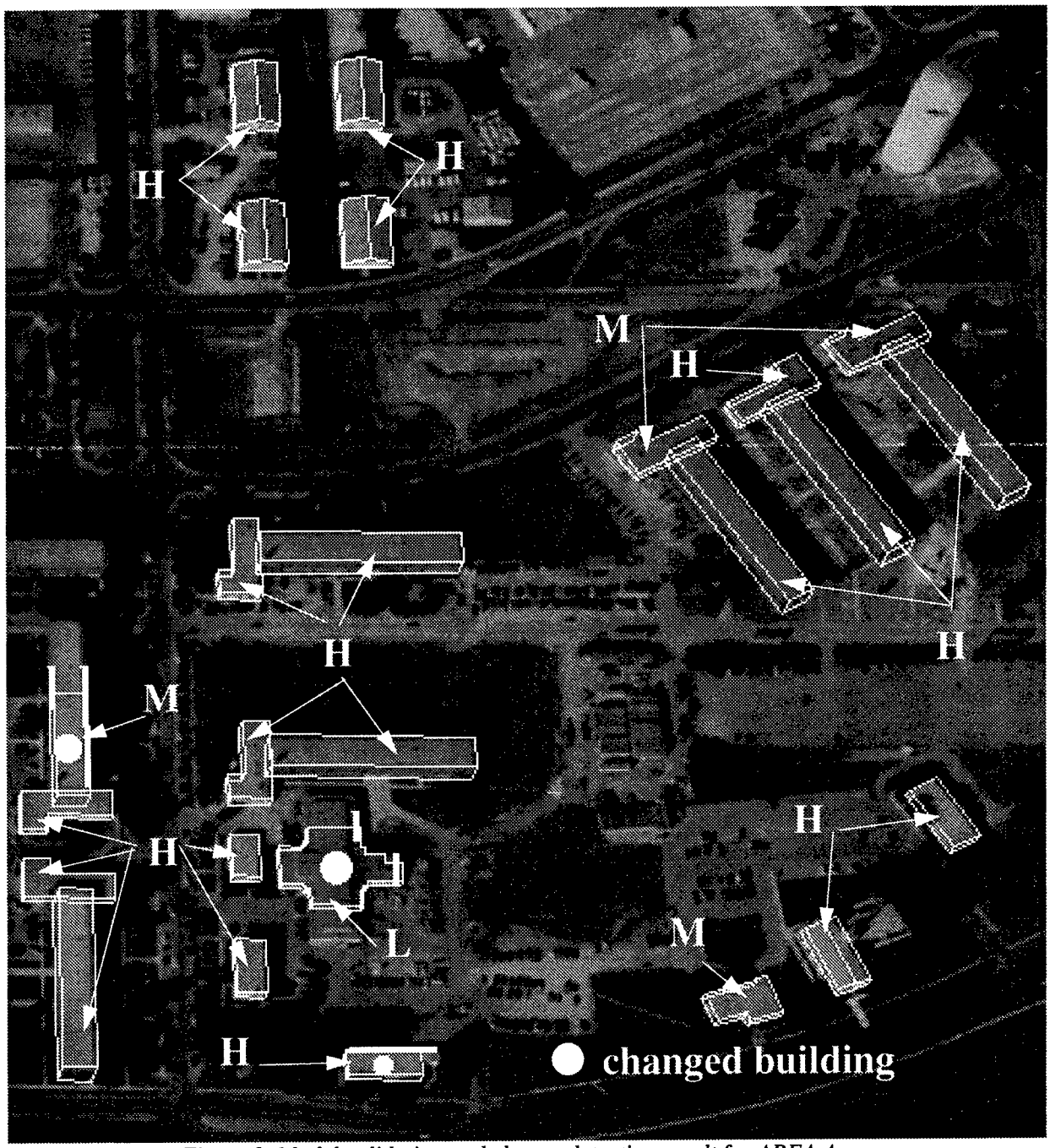


Figure 3 Model validation and change detection result for AREA A

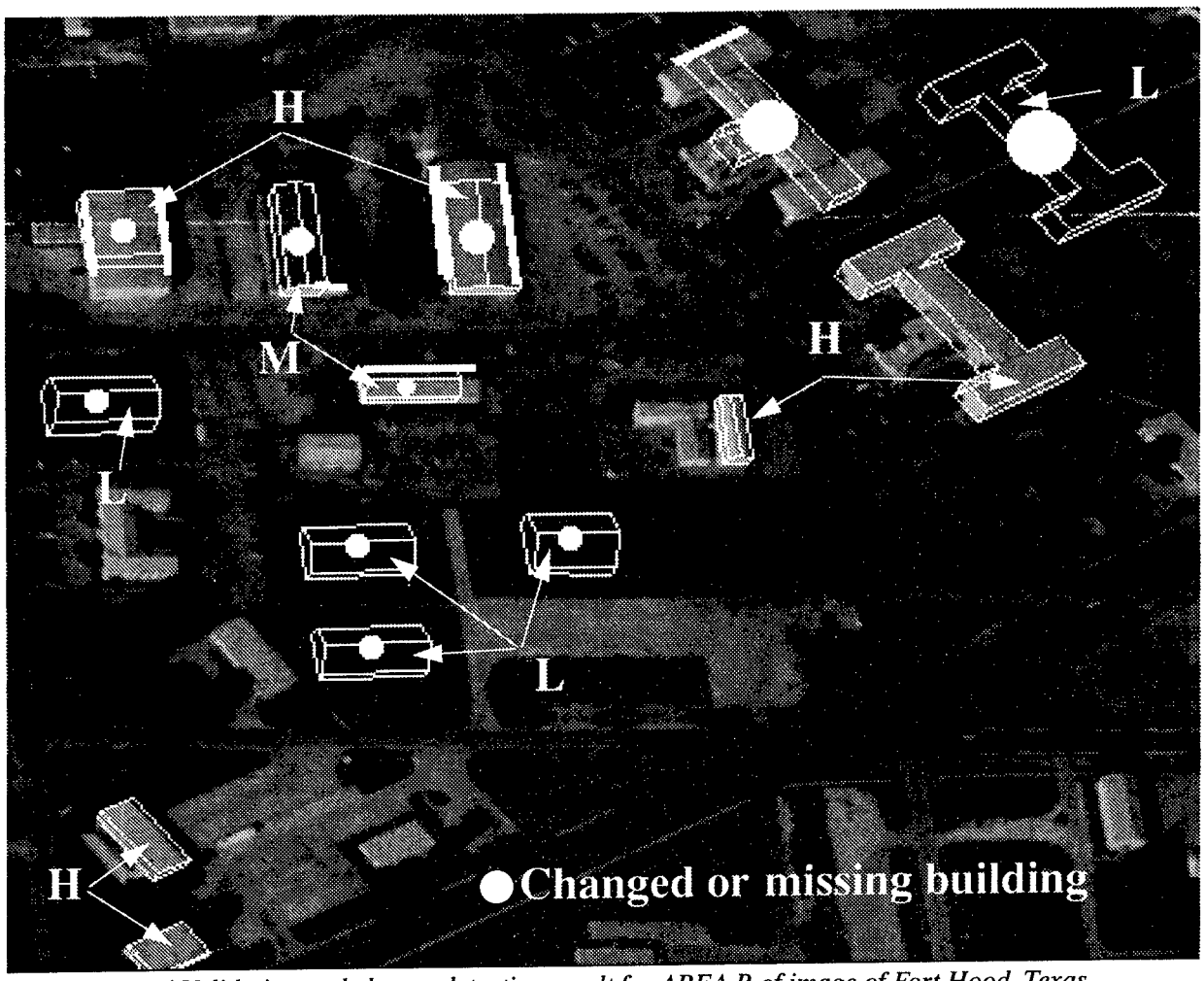


Figure 4 Validation and change detection result for AREA B of image of Fort Hood, Texas

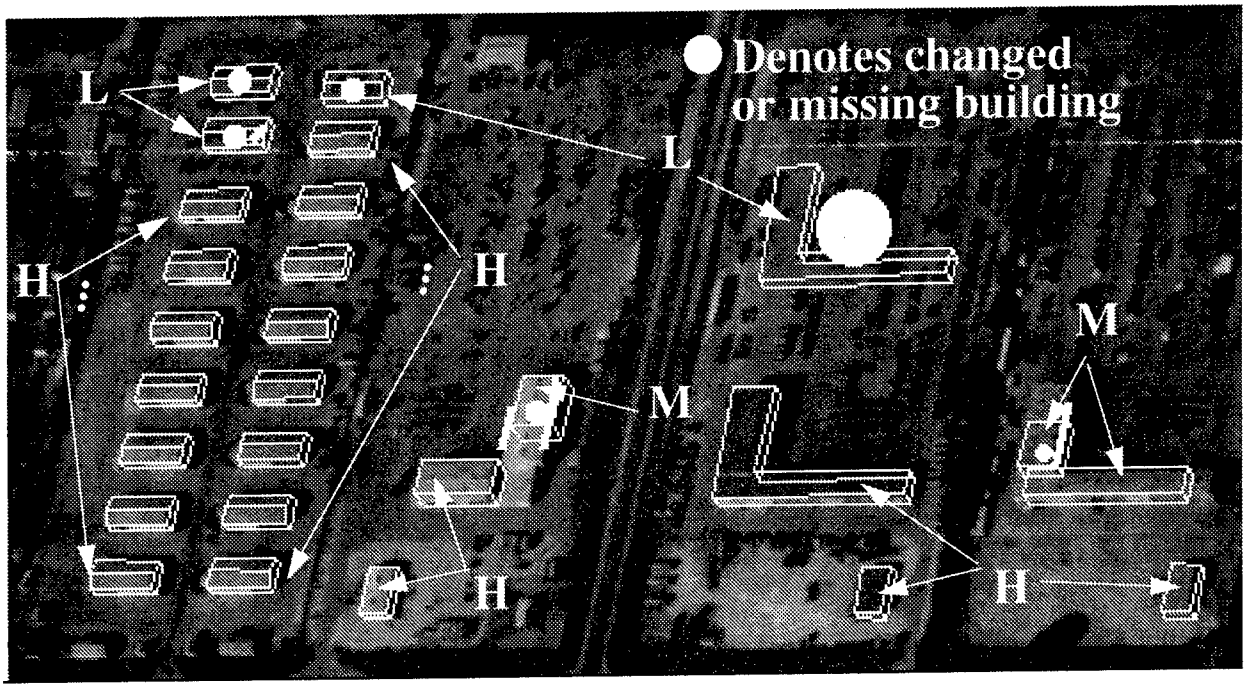


Figure 5 Validation and change detection result for AREA C of image of Fort Hood, Texas

buildings, 54 were modeled to reflect the underlying structure, 14 were modeled with changes in their dimensions, and 11 missing (in the model but not in the site were added. In this experiment, only one non-changed building is reported as changed due to a coincidental alignment not currently considered by the system; one changed building was not reported as changed, and all missing buildings are found to be missing in the image.

One important type of site change is the introduction of new structures. We have capabilities to construct models automatically and therefore we can suggest new additions to the site model (see Sections 3, 4 and 5 below.) These techniques are applied to areas of interest, currently designated in the site model as "functional areas", using one or more images, if available. The site model is used to indicate already modeled areas.The camera models and terrain models associated with the images are used also by these systems to derive viewpoint and illumination parameters automatically. An example of this task is shown in Figure 6. In this experiment we removed the three buildings from the model in the lower right part of AREA C. The model construction system is cued to detect new structures there and add these to the site model.

Our system has been tested largely on real images of the Ft. Hood, TX site. Generally, the performance of the registration, validation and change detection system is very robust. Some changes may be more apparent in other views, or may need to be confirmed using other views.

Table 1: Summary of Results

Image (fhov927)	Visible Buildings	Non-changed Buildings			Changed Buildings			Missing Buildings		
		Number of buildings	Reported non-changed	Reported changed	Number of buildings	Reported changed	Reported non-changed	Number of buildings	Reported missing	Validated
No.	79	54	53	1	14	13	1	11	11	0
%	100.0	100.0	98.1	1.9	100.0	92.8	7.3	100.0	100.0	0.0

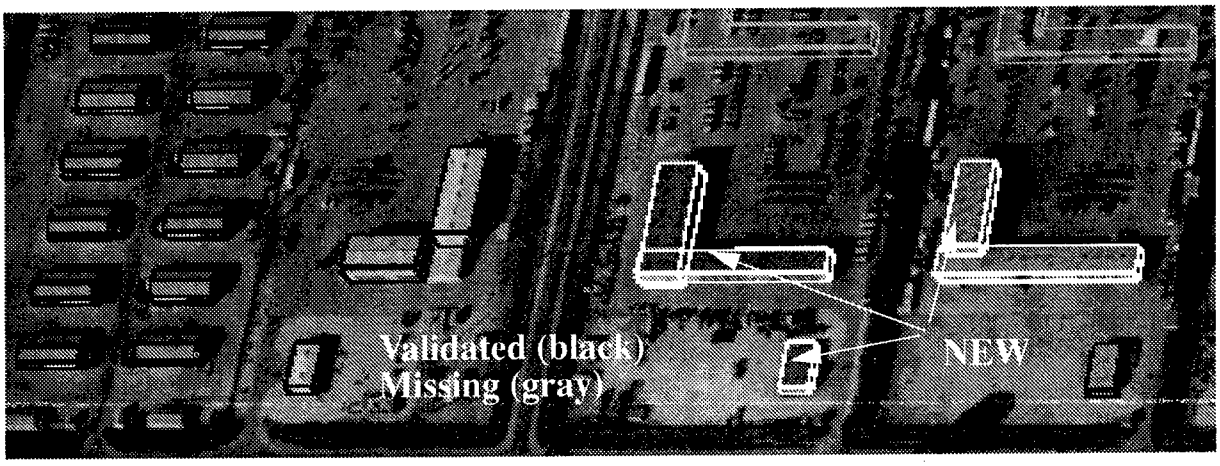


Figure 6Model model updating: New buildings are detected automatically.

3 Automated Building Detection and Description

Buildings are the dominant 3-D stationary structures in most sites. It is important to automatically detect and describe them, for initial site model construction as well as for change detection and site model updating. We have developed two systems for this task: one that processes single intensity images separately and then combines the results from the various views, and another that uses a number of images concurrently during the process. It is, of course, easier to detect and describe buildings using multiple images, however, the ability to at least reliably detect buildings from a single image (monocular system) is needed during the change detection process and the system does not require a very accurate camera model.

Automatic building detection and description is difficult due to several reasons. Images of outdoor sites are quite complex. Roofs of buildings are not necessarily homogeneous nor are their boundaries always well delineated from the background. Furthermore, the background itself contains many features such as roads, vehicles, landscaping and other vegetation. This makes the process of segmentation difficult and requires use of sophisticated perceptual grouping methods. The images also lack direct 3-D information. In monocular analysis, we infer heights from shadows and from projected lenghts of walls (if they are visible). When multiple images are available, heights can be inferred by matching features in the multiple images. However, this can only give us sparse 3-D data and the matching of features themselves can be quite ambiguous as many similar features may be present nearby.

We have made good progress on both the monocular and the multiple image systems. Both are currently restricted to analyzing rectilinear structures and assume that camera models and the sun positions are known.

3.1 Integration of Monocular Results

The monocular system has been described in the literature [Lin et. al. 1994, Lin and Nevatia 1996]. Here we give an overview of the recent progress in integrating the results from multiple views. The system can integrate the results from multiple views by projecting the hypotheses of one view into the other views and verifying them in all views. Assume that a set of images are taken from several different viewpoints. The evidence of a building may be more clear in one view than in the others, depending on several conditions, such as the viewing direction, the illumination direction, and the building orientation. Once a building can be correctly detected by the system in one view, it is very likely that the system can find supporting evidence of this building in other views. On the other hand, if the system makes an incorrect hypothesis in one view, it would be very unlikely for the system to find much supporting evidence from other views. Based on this observation, the system can make a better decision by integrating all evidence from all available views. Here we assume that accurate camera models are given to the system for registration.

This process is different from the traditional stereo techniques. It is a top-down process which analyzes the underlying evidence of a high level hypothesis instead of trying to match low level features, such as lines and junctions, between views. The success of this process highly depends on the results of the monocular system. Especially the hypothesis generation process must be able to create appropriate hypotheses, which are the basic entities of the integration process.

First, the system detects buildings from each image individually. Assume that the camera model for each view is given. The system can project the 3-D wire frame of a verified hypothesis in one view into another view. All evidence around the projected wire frame of the verified hypothesis in the second view is collected and then the evaluation function of the hypothesis verification process is applied on the collected evidence to compute the confidence of the hypothesis in this view. The confidence values of a hypothesis in all views are combined using principles of certainty theory. It is possible that a verified hypothesis in one view has negative confidence on another view. A threshold value depending on the number of views is set to remove those unsatisfactory hypotheses.

Another problem of combining results from multiple views is that a building could be verified individually in more than one view. Multiple hypotheses could be retained for a single building after the thresholding. Therefore, an overlap analysis is required to compare the combined confidences of those hypotheses overlapped in 3-D space. The hy-

pothesis with the highest combined confidence is retained and the rest overlapping hypotheses are removed. Finally, a set of 3-D cube features is created for the list of retained hypotheses. The user can examine the results by projecting these 3-D cube features into any of the views.

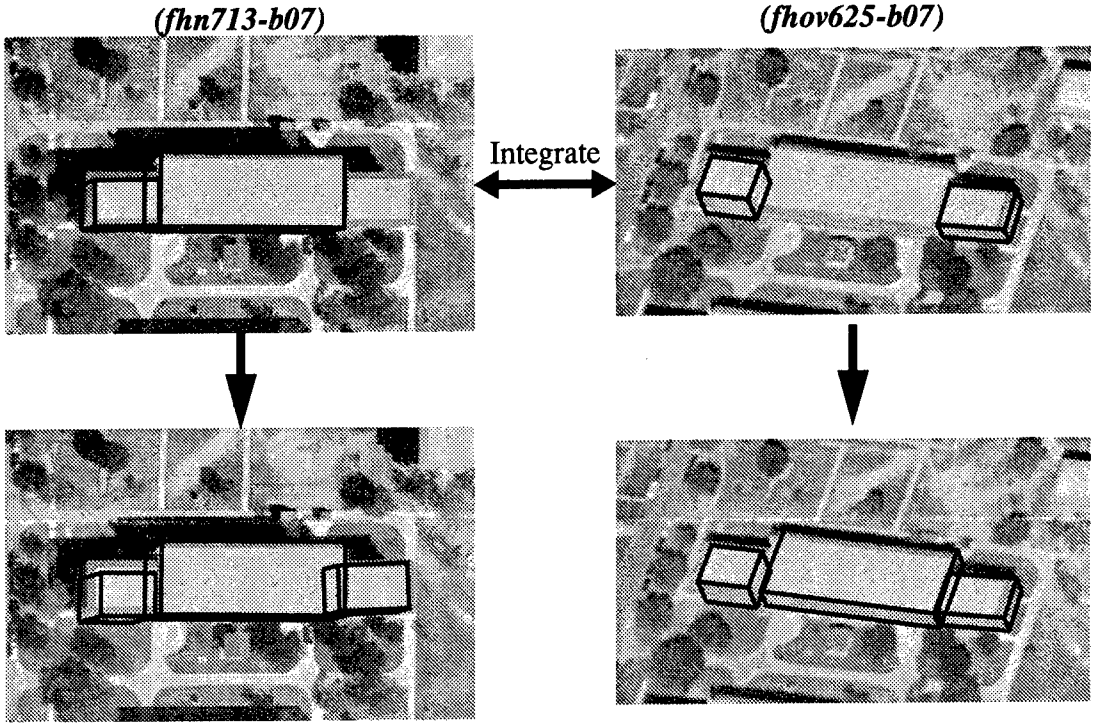


Figure 7 Integration of results from multiple views.

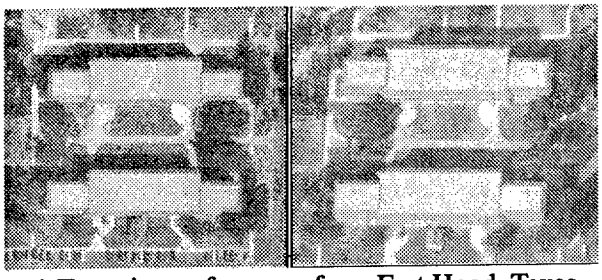
A more sophisticated algorithm is required when none of the hypotheses of a building from all views is correct. The algorithm must be able to fuse the evidence of the building from different parts of those hypotheses and create the correct hypothesis. Some part of the evidence of a building will be stronger in one view than the other depending on the situations, such viewpoint direction and illumination direction. The algorithm can decide how to combine the evidence based on the situations of all views.

Figure 7 shows an example of integrating the results from two views of a building. The building is composed of three structures. The main structure in the middle is detected from the image, FHN713-b07. The right wing of the building is detected from the image, FHOV625-b07. The left wing of the building is detected on both images. One of the two hypotheses, corresponding to the left wing of the building, therefore must be selected based on the combined confidence of the hypotheses. After the integration process, the three structures of the building are all verified, and the projections of the 3-D results are shown at the bottom of Figure 7.

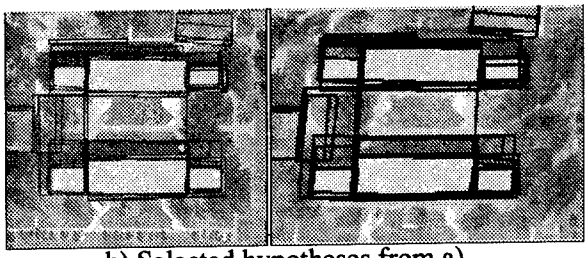
3.2 Multi-View System

Our system using multiple images uses a hierarchical grouping and matching methodology which generate roof hypotheses. These are again verified by shadow and wall evidence, if available. Information from all views is used in a non-preferential way. The preliminary results are encouraging and we believe that this method will lead to robust and reliable building detection and description. An example of the results of this system on segments of the modelboard images is shown in Figure 8. Thirteen of the sixteen buildings in this example are detected. The missing buildings are, as in the monocular system, either small or have dark roofs. There are no false positives. This system also generates a confidence value with each result.

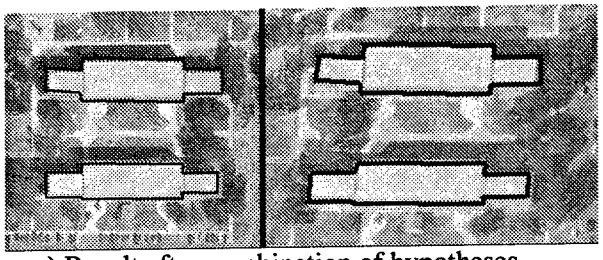
Testing of this system continues and we are getting encouraging results on the Fort Hood images. Details of this system are given in [Noronha and Neva-



a) Two views of a scene from Fort Hood, Texas



b) Selected hypotheses from a)



c) Result after combination of hypotheses

Figure 8 Multi-view building detection

tia 1997].

4 Interaction with Automatic Model Construction Systems

Even though the performance of automatic building detection systems is improving rapidly, the results are not perfect. We have been developing a system for user interaction with our automated systems to allow for easy correction or completion of the results of the automated analysis. Our system is based on our observation that our automated systems fail under certain conditions for certain specific reasons. If these reasons are known and some guidance is provided to the automated system, it can then complete the task.

For example, our monocular building detection fails to find buildings with dark roofs in some cases as the roof boundary can not be distinguished from the adjoining shadow. The system does make some hypotheses for the presence of a roof but none are judged to be strong enough to qualify as a validated result. In such cases, it is sufficient for the user to indicate the cause of failure to the system and to point somewhere in the interior of the undetected roof. With this minimal interaction, the system can find the previously undetected building. In some cases, more extensive interaction may be needed and the user may need to correct the position or size of the detected roof. However, even here, the user only needs to make some corrections, the rest is done automatically. For example, moving the corner of a roof will result in the system automatically determining a new height for the roof. Details of the earlier capabilities of this system are given in [Heuel and Nevatia 1995].

Here, we give an overview of recent additions to the system and give some preliminary performance results and comparisons of time and effort required. We have incorporated an "assisted" mode into the system. The user assists the system by indicating by mouse clicks the location of a building roofs in the image. The system collects these inputs and proceeds with automatic detection as usual, to give roof hypotheses at those locations. The calculations of confidence levels based on underlying image support for the roof and shadow and wall evidence are collected as well. Next we show an example of the results that can be expected from this kind of interaction.

Figure 9 shows a projection of the 3-D model computed automatically with no assistance on a portion of an image from the Fort Hood site. As pointed out above, the automated result is quite good although some buildings are not described. Figure 10 shows the 3-D model constructed from hypotheses returned in assisted mode. The result requires additional editing of some of the roofs. These however, require minimal interaction as the editing and recalculation of confidence values and collection of shadow and wall evidence is carried out automatically by the system. The final result is shown in Figure 11.

The amount of effort in time and in labor to generate this result is summarized in Table 2. The comparison is with the effort required to produce the same result by hand using traditional modeling tools, such as those supplied with the RCDE [Strat et al, 1992], and those including assistance and minimal user interaction. The figures corresponding to the example above are in the table rows labeled "complex". In Table 2, t_m, t_i, and t_e denote time in minutes for manual, interactive and editing process-

es respectively. The speed-up in time is by a factor of 7.2 for the "complex" example shown. As shown on the table, for the three types of shapes, the speed-up increases as the shape complexity is reduced. These results are preliminary and extensive testing is needed to quantify its utility. The ability to construct complete 3-D models of sites with buildings automatically is the desirable goal. With minimal interaction however, it becomes possible to construct these models, mostly automatically, today.

5 Conclusion

We have summarized our research activities in analysis of overhead images. These consist of change detection, site model construction and site model updating. We believe that many of these techniques are ready to be transitioned for testing in operational environments. We are in the process of running more extensive tests and of porting the software to operational environments.

Acknowledgments

This paper describes work of Chungan Lin and Sanjay Noronha.

Table 2: Time Comparison (time in minutes)

Image Description	# of Buildings	# of Boxes	t_m	t_i	t_e	$t_i + t_e$	# of Boxes edited	$\frac{t_m}{t_i + t_e}$
L-shape	8	12	8	0.2	0.5	0.7	2	11.4
I-shape	19	35	28	0.6	2.5	3.1	4	9.0
Complex	14	27	75	0.4	10	10.4	7	7.2

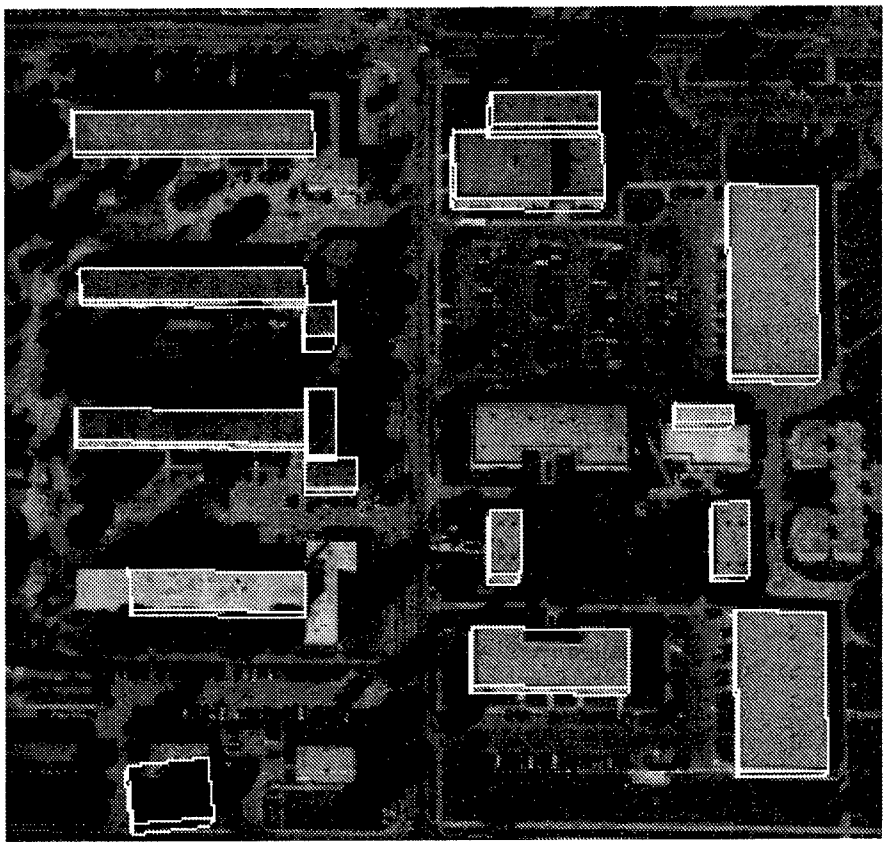


Figure 9 Results of automatic processing

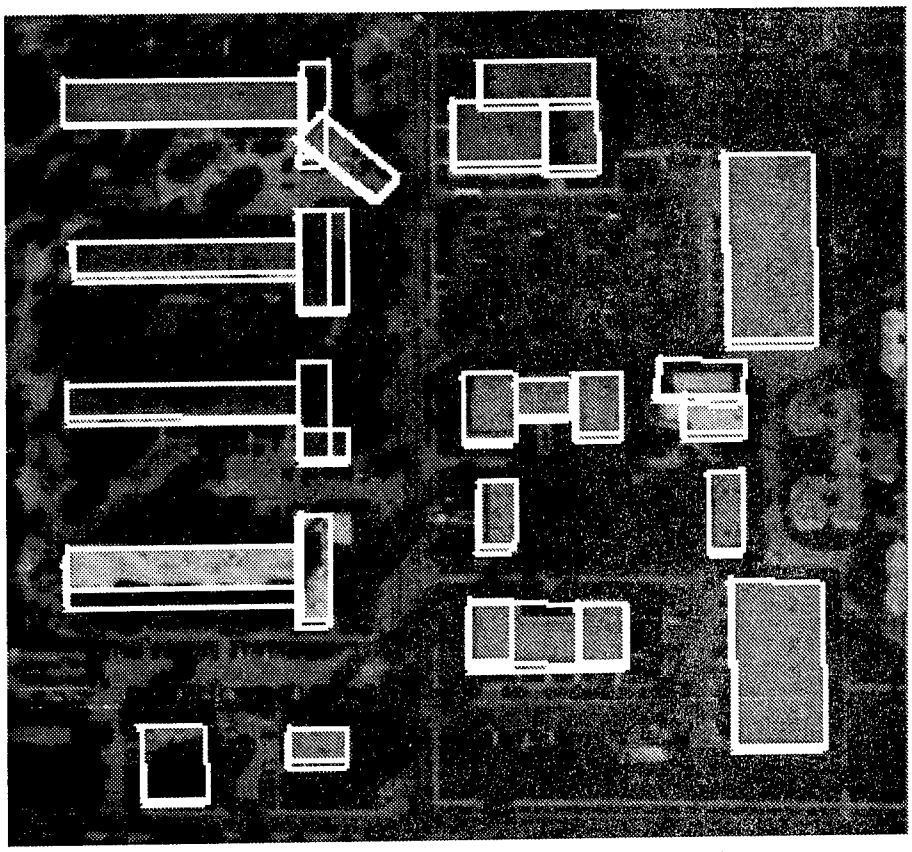


Figure 10 Results of automated assisted processing

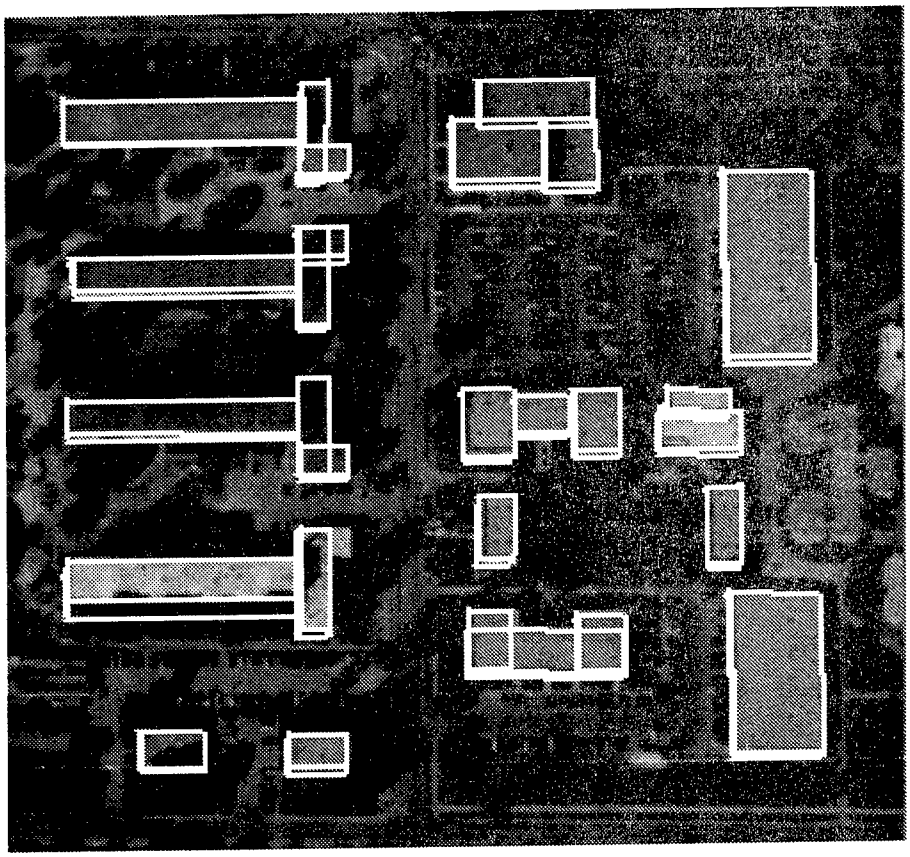


Figure 11 Final results after minimal interaction

References

[Gerson and Wood 1994] Gerson, D. and Wood, S., *"RADIUS Phase II. The RADIUS Testbed System,"* Proceedings of the Image Understanding Workshop, Vol1, Monterey, California, November, pp. 231-237.

[Huertas et. al. 1995] Huertas, A., Bejanin, M. and Nevatia, R., *"Model Registration and Validation,"* In Automatic Extraction of Man-Made Objects from Aerial and Space Images, Gruen, A., Kuebler, O., Agouris, P. Editors. Birkhauser Verlag, Switzerland, pp 33-42.

[Huertas and Nevatia 1997] Huertas, A. and Nevatia, R., *"Model-Based Change Detection in Man-Made Structures."* in Proceedings of the DARPA Image Understanding Workshop, New Orleans, Louisiana, May, these proceedings.

[Lin et. al. 1994] Lin C., Huertas A. and Nevatia R., *"Detection of Buildings using Perceptual Grouping and Shadows."* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, pp 62-69.

[Lin and Nevatia 1996] Lin, C. and Nevatia, R., *"Building Detection and Description from Monocular Aerial Images,"* in Proceedings of the DARPA Image Understanding Workshop, Palm Springs, California, February, pp 461-468.

[Noronha and Nevatia 1997] Noronha, S. and Nevatia, R., *"Detection and Description of Buildings from Multiple Aerial Images,"* in Proceedings of the DARPA Image Understanding Workshop, New Orleans, Louisiana, May, these proceedings.

[Marouani et. al. 1995] Marouani, S., Huertas, A. and Medioni, G.), *"Model-Based Aircraft Recognition in Perspective Aerial Imagery,"* Proceedings of the IEEE Symposium on Computer Vision," Coral Gables, Florida, November, pp. 371-376.

[Heuel and Nevatia 1995] Heuel, S. and Nevatia, R., *"Including Interaction in an Automated Modeling System,"* Proceedings of the IEEE Symposium on Computer Vision," Coral Gables, Florida, November, pp. 383-388.

Strat et al, 1992] Strat, T. et al., *The RADIUS Common Development Environment*, Proceedings of the DARPA Image Understanding Workshop, San Diego, California, Morgan Kaufman, Publisher, January, pp 215-226.

A Real-Time, Interactive SAR Tactical Mapper

John B. Hampshire II

Institute for Complex Engineered Systems (ICES)
and
Department of Electrical & Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Email: hamps@ece.cmu.edu
http://gussolomon.ius.cs.cmu.edu/hamps/IU/index.html

Abstract

This report describes sarMapper, a newly-funded research project to develop real-time, interactive software for generating high-accuracy tactical maps from synthetic aperture radar (SAR) imagery. High accuracy ground cover maps make it possible to automate the focus-of-attention mechanism that is the foundation of tactical image analysis. The sarMapper *development* goal is to allow a single human intelligence officer to monitor a tactical area on the order of hundreds to thousands of square kilometers, looking for vehicles, roads, construction sites, and structures — any man-made ground cover — of potential tactical interest. The sarMapper *research* goal is to create efficient, semi-autonomous algorithms that realize the development goal in real-time on a current-generation laptop computer. This document outlines the sarMapper project goals for calendar year 1997.

This newly-funded research is sponsored by the Defense Advanced Projects Research Agency under grant F33615-91-1-1017, monitored by the United States Air Force Wright Laboratory ATR Development Branch, Wright Patterson AFB, Dayton, OH. The views and conclusions contained in this document are the author's and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Project Agency, Wright Laboratory, the U.S. Air Force, or the U.S. government.

1 Introduction

The sarMapper project seeks to generate high-accuracy ground cover maps from SAR imagery and to use those maps to target locales within a wide search area for further human and/or machine analysis. In short, sarMapper will provide the military with a tactical map generation and focus-of-attention capability that will run in real-time on an off-the-shelf laptop computer.

The sarMapper concept is based on the following operational notions:

- military image analysts are drowning in a sea of data for lack of a real-time ability to process the data into usable information.
- military commanders need accurate tactical maps *now*, not two hours from now.
- they need them in the field, where their troops can use them; the troops need to be able to update their maps rapidly in order to reflect the changing tactical situation in real-time.
- they must be able to do this with minimal effort and training and little or no prior information regarding the area being imaged by the SAR reconnaissance platform.

Likewise, the sarMapper concept is based on the

following technical notions:

- a little human oversight can go a long way towards building a nearly autonomous tactical mapper.
- objects of tactical interest are generally man-made; modern man-made materials generally have radio-frequency (RF) backscatter signatures that are distinctly different from those of natural objects.
- searching for objects of tactical interest is rather like winnowing wheat from chaff — first we separate the wheat from the chaff and then we decided which kernels of wheat are fit to eat
- likewise, objects of tactical interest can be detected efficiently over a wide area by first mapping that area to determine where there is man-made ground cover and then focussing attention on the man-made areas to see if they constitute objects of tactical interest.

This, then, is the purpose of sarMapper: to generate high-accuracy tactical maps in real-time and focus the attention of human and/or machine analysts on man-made regions of the map.

2 Objectives

The sarMapper project seeks to provide the military with two fundamental capabilities:

- Fast, automated, high-accuracy, tactical map generation.
- Tactical focus of attention.

sarMapper itself is to have the following characteristics:

- Real-time learning & map generation
- Without prior ground-truth
- On a laptop (with a CD-ROM or large external disk)
- Computational efficiency (generate a high-accuracy mega-pixel map in one to three minutes on a laptop)
- High resolution / accuracy ground cover assessment
- Mega-pixel map in a minute
- Interactive graphical user interface (GUI): human aids computer in initial learning phase; after learning, computer maps autonomously
- Human oversight
  - · Assess focus-of-attention warnings
  - · During sarMapper's learning phases
- Multiple
  - · Wavelength (P-, L-, C-, X-Band, etc.)
  - · Polarization (single and fully polarimetric)
  - · Spatial resolution
  - · Data Sources (e.g.,...)
    - – MSTAR public data (airborne, X-band)
    - – LL ADTS SAR (airborne, X-band)
    - – JPL AirSAR (airborne, P, L, and C-band)
    - – NASA SIR-C/X-SAR (spaceborne L, C, and X-band)
- Focus-of-Attention (FOA)
  - · Detect & locate man-made ground cover
  - · Warn human according to prior tasking
- Usable with ~1 hour training

Computational efficiency forms the core of sarMapper, allowing it to generate mega-pixel maps on a current-generation laptop in one to three minutes. Ground cover types are learned using low-complexity parametric models of RF backscatter: learning takes the form of efficient model parameter estimation, which allows ground cover types to be characterized in terms of their backscatter signature — the learning and subsequent mapping take place in real-time, owing to the efficient, low-complexity algorithms employed. In comparison, standard maximum-likelihood map generation algorithms generate maps on a time scale of hours.

sarMapper uses *semi-supervised* learning, which obviates the need for prior ground truth. The principle behind this supervised learning procedure is straightforward: humans can discern different ground cover types in a SAR image by the differences in their appearance in the image. Different ground cover types in a single-polarization image will appear to have different shades and/or textures of gray; in a false-color composite of multiple polarizations, different ground cover types will appear in different colors. Consequently, a human can identify regions of different ground cover in an image and label these regions without knowing what the different ground cover types are — using pseudonyms for the unknown ground cover classes. These pseudo-classes of ground cover can be learned, and their backscatter signatures can then be used to generate a high-resolution pseudo-map over a wide-area in the vicinity of

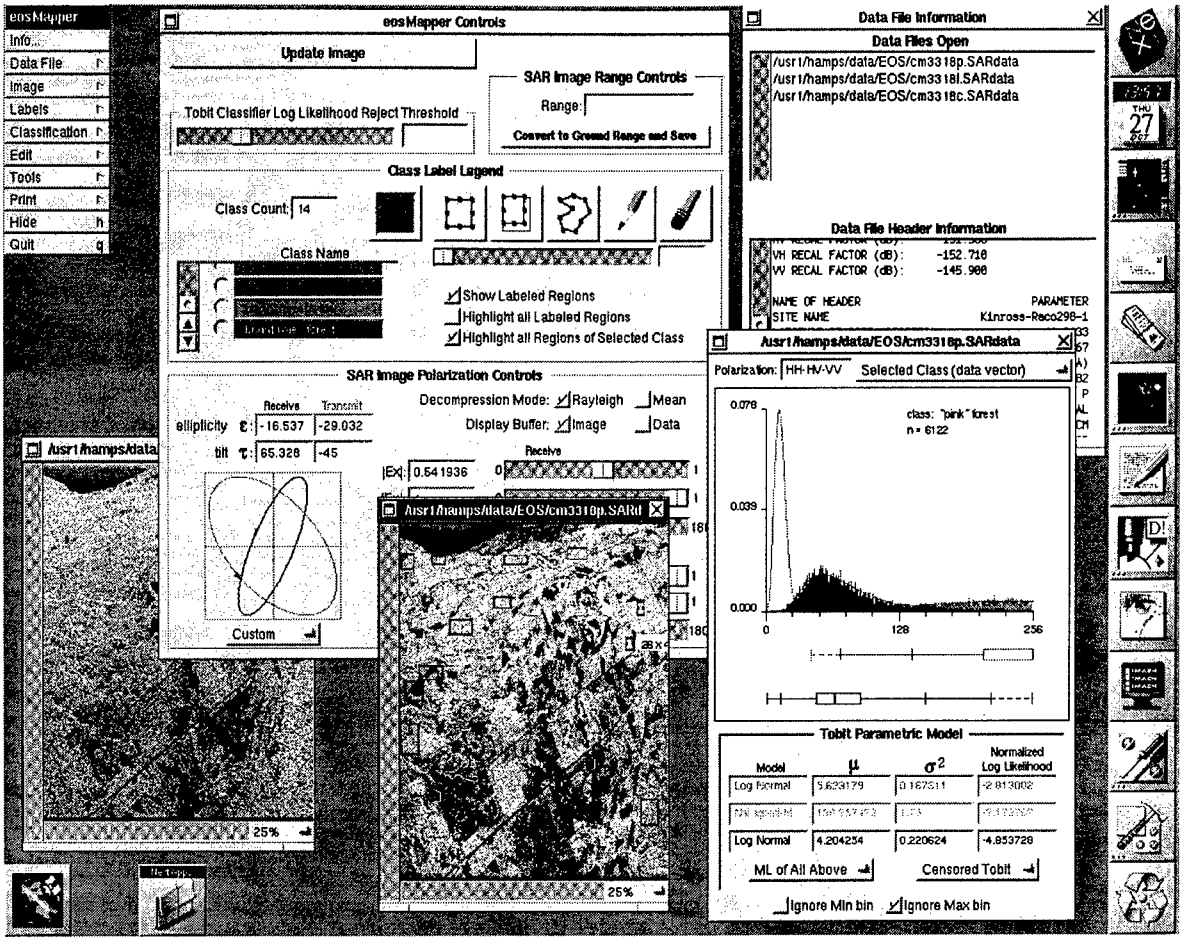


Figure 1: The sarMapper graphical user interface (GUI). The main controls allow the human operator to specify the polarization(s) of the SAR image, the number and type of ground cover classes, image labeling modes, etc. Multiple images can be loaded and displayed simultaneously (HH-HV-VV composites are shown for C- and P-band images of NASA's Raco, Michigan super site imaged by the JPL AirSAR platform). Histograms of the backscattered SAR radio frequency (RF) envelope are used to derive parametric models for each ground cover class. These ground cover "signatures" are in turn used to produce a ground cover map for the site and surrounding areas. Because the signatures are derived from 8-bit representations of the SAR backscatter, automatic ground cover learning and subsequent mapping can be done in near-real time. sarMapper's speed and interactive GUI make it well suited to automatic wide-area tactical monitoring and focus-of-attention tasks.

the image used for learning. Many of the unknown ground cover types can be inferred by an image analyst from context, site-invariant backscatter signatures, historical imagery, or focussed follow-on surveys conducted using the pseudo-map to target specific survey sites. The critical characteristic of semi-supervised learning is that it can generate a useful map in real-time without prior knowledge of the area; missing details can be filled in as they are obtained, without having to re-learn or re-map the area.

Since man-made ground cover tends to backscatter little RF energy (as in the case of obliquely illuminated metal or concrete surfaces) or substantial RF energy (as in the case of trihedral reflectors common to military vehicles), semi-supervised learning is compatible with sarMapper's focus-of-attention (FOA) mission. Very large areas (hundreds to thousands of square kilometers) can be mapped and surveyed for small areas of potential tactical interest by a

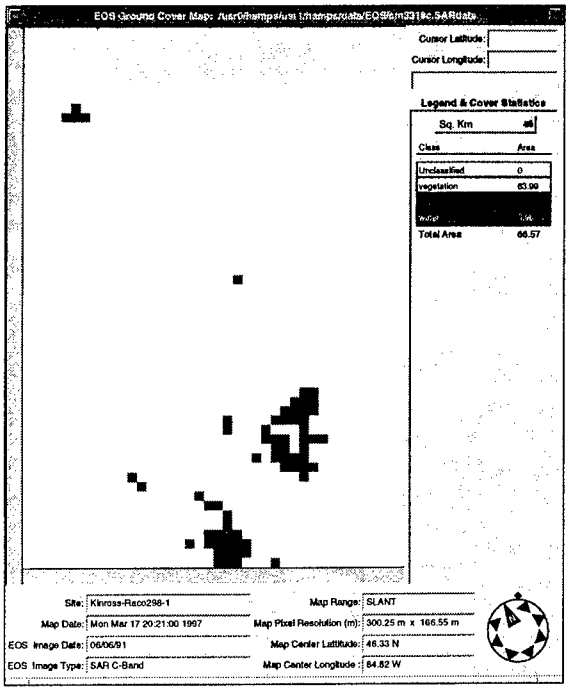


Figure 2: A coarse(300 m) resolution focus-of-attention map of the Raco, Michigan site, generate from the partially obscured C-band image in figure 1 (HH-polarized backscatter only). Darker regions on the map indicate areas of potential interest (water or tarmac; the latter is of tactical interest). sarMapper generated this map in five seconds. High-resolution mapping and Automatic Target Recognition can be focussed on these areas.

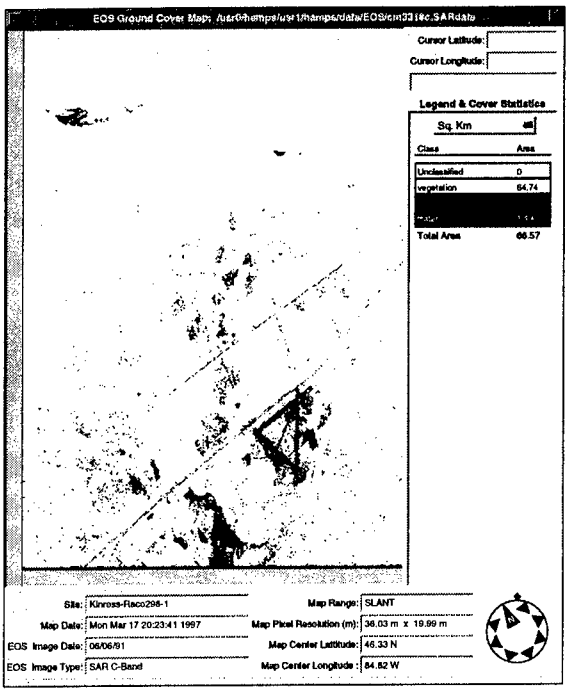


Figure 3: A fine(36 m) resolution map of the Raco, Michigan site, generate from the same C-band HH-polarized image used to generate the focus-of-attention map in figure 2. sarMapper generated this map in 20 seconds. The darkest regions on the map indicate areas likely to be tarmac. The physical structure of the airfield, combined with its tarmac ground cover identify it as the primary target of interest in this 67 square-kilometer area.

single human and sarMapper team. Survey time for a thousand square kilometer area (10-meter map resolution) requires between ten minutes and one-hour on a single laptop computer, depending on the number of ground cover classes enumerated; survey time for a fifty square kilometer area requires between 15 seconds and three minutes under the same conditions.

3 Preliminary Results

Figure 1 illustrates the sarMapper graphical user interface (GUI) . The interface allows a human operator to view multiple SAR images simultaneously. Fully-polarimetric SAR images can be displayed in HH-HV-VV false color composites: two such images (one C-band and one P-band) are shown for NASA's Raco, Michigan site (JPL AirSAR imagery). The human operator uses the GUI to identify areas of distinct backscatter signature; these areas correspond to different looking regions in the image, which , in turn, correspond to different types of ground cover. By identifying different ground cover types using a simple GUI, the human operator provides sarMapper with the information it needs to learn quantitative radio-frequency (RF) backscatter ''signatures'' for each ground cover class. A detailed description of this semi-supervised learning procedure is described in [Hampshire-97].

Once sarMapper has learned the RF backscatter signatures that characterize the different ground cover classes, it can generate a ground cover map of the imaged area — and all the surrounding areas in which the ground cover is similar. Because sarMapper's mapping speed is approximately four orders of magnitude faster

than conventional maximum-likelihood SAR mapping algorithms [Hampshire-97] it can generate a high-resolution mega-pixel map in one to three minutes on a current technology laptop, depending on the level of map detail required. sarMapper generated the low-resolution map of figure 2 in five seconds on a SPARC 20 workstation; it generated the medium resolution map of figure 3 in 20 seconds. Mapping speeds for current-technology laptop computers are approximately equal.

Figures 2 and 3 illustrate how sarMapper can be used to generate ground cover maps that focus the attention of a human SAR analyst monitoring (potentially) thousands of square kilometers of territory. Figure 2 shows the result of a low-resolution first-pass analysis of the Raco, Michigan area. This map shows seven areas of potential man-made material, displayed as dark pixels on a lighter background (dark pixels belong to the same larger area of interest if they are separated by no more than one light pixel). Using figure 2 as an overlay mask on the map in figure 3, sarMapper identifies three areas that might contain large amounts of tarmac: of these three areas, two lack man-made geometry (indicating water, which can exhibit a backscatter signature similar to tarmac). The third area is an airfield with obvious man-made geometry.

4 Research Questions

The initial prototype of sarMapper described in the preceding section instantiates the semi-supervised learning and efficient map generation algorithms described above and is described in detail in [Hampshire-97]. This prototype has been subjectively evaluated using long-wavelength (P, L, and C-band) SAR imagery from the JPL AirSAR platform.

Four research questions will be addressed this year. These questions — all of them technical in nature — touch on the speed, accuracy, and robustness of sarMapper's mapping and focus-of-attention capabilities:

- Can high-accuracy maps be generated from 8-bit X-band backscatter envelope data?
- Is semi-supervised learning a statistically consistent paradigm?
- Can man-made ground cover be identified consistently in SAR imagery and pseudo-maps without ground truth?
- How can a robust focus-of-attention algorithm be derived for real-time laptop implementation?

5 Evaluation

Pursuant to research efforts to address them, the research questions listed in the previous section will be answered by objective evaluation of sarMapper using SAR data from a wide variety of sensors. Pre-processing will be added to sarMapper so that it can map imagery from these three sensors (in addition to the JPL AirSAR platform):

- Lincoln Lab ADTS SAR (X-band)
- MSTAR SAR (X-band)
- NASA SIR-C/X-SAR (L, C, X-band)

Speed: Average semi-supervised learning time will be assessed with human-computer timing trials. Map generation times will be tabulated for a corpus of evaluation images.

Accuracy: Map accuracy will be assessed using imagery for which ground truth is known or can be infered. Accuracy will be quoted according to general methods of statistical inference/pattern recognition, with 95% confidence bounds and ground cover confusion matrices derived from test images (or test areas within an image) not used during semi-supervised learning.

FOA: sarMapper's focus-of-attention algorithm will be assessed according to general methods for evaluating detection algorithms; namely, receiver operator characteristic (ROC) curves will be generated and evaluated for FOA performed on test imagery/maps not used for semi-supervised learning.

References

[Hampshire-97] J. B. Hampshire II, *sarMapper: A real-time, interactive SAR tactical mapper*, Proceedings of the 1997 DARPA Image Understanding Workshop, *to appear*, May, 1997.

Image Understanding at Lockheed Martin Valley Forge

Anthony Hoogs, Doug Hackett and Tom Barrett
Lockheed Martin Management and Data Systems
P.O. Box 8048
Philadelphia, PA 19101
[hoogs|hackett|barrett]@mds.lmco.com

Abstract

The image understanding group at Lockheed Martin Management and Data Systems is engaged in a range of IU-related projects. The recently-completed RADIUS program successfully demonstrated a number of IU technologies that can be applied to imagery analysis in the near future. As the prime contractor for RADIUS, we were responsible for the RADIUS Testbed, including the integration of IU algorithms from a number of institutions. The technologies pioneered in RADIUS are being transitioned to operational prototypes in the Site Monitoring System and the Spatial Image Annotation System, both of which use model-supported exploitation as a central framework. In addition, we have pursued corporate-sponsored, basic IU research in object representations and change detection, developing new techniques to exploit site model context in automated imagery analysis.

1 Introduction

The Image Understanding group at Lockheed Martin Management and Data Systems is involved with a number of projects ranging from basic research to the production of near-operational software.

While Management and Data Systems focuses on producing very large, operational software systems, our group concentrates on state-of-the-art technology development through technology contracts and independent research and development (IR&D). We are heavily involved with the transfer of IU technology into near-operational and operational domains, but we also keep current in IU technology through basic IU research.

Our recent government-sponsored projects include Research and Development for Image Understanding Systems (RADIUS), the Site Monitoring System (SMS), and the Spatial Image Annotation System. RADIUS focused on IU technology development and integration, with a significant effort toward development of a prototype workstation. Currently in progress, the SMS will transfer selected parts of RADIUS to an operational prototype workstation for performing semi-automated, tactical site monitoring under the larger Semi-Automated IMINT Processing (SAIP) Advanced Concept Technology Demonstration sponsored by DARPA. SIAS is a smaller, short-term effort intended to package a compact subset of RADIUS workstation capabilities in a model-supported exploitation workstation.

Our corporate-sponsored research is aimed at developing algorithms that fully exploit site model context to increase robustness. As a foundation for this goal, we have pursued research in object representations that combine geometry and photometry to provide more accurate appearance modeling of visually complex features. We have applied these techniques

to change detection and pose refinement algorithms within the site model framework.

These efforts are described in the following sections, although the RADIUS project and our research results are detailed elsewhere in these proceedings.

In addition to the authors, the IU group at LMC M&DS includes Bill Bremner, Mark Horwedel, Mike Puscar, Tony Canike, Norris Heintzelman, Mark Thompson, David Dadd, Mike Lentowski, Bill Brooks, and Jim Kennedy.

2 RADIUS

The Research and Development for Image Understanding Systems (RADIUS) program has been an ongoing effort for the last six years. RADIUS Phase II, which focused on the development of the RADIUS Testbed System (RTS), will be completed in early 1997.

RADIUS Phase II culminated in the Baseline Delivery in July 1996. At that point the RTS became a fully developed, prototype model-supported exploitation workstation. Using the RTS, an analyst can construct site models using manual, semiautomated, and automated tools. Model-supported exploitation (MSE) can be accomplished with a large suite of tools supporting feature-specific data access, historical queries, image/site model queries, and, most significantly, automated exploitation. A detailed summary of the Phase II program is given elsewhere in these proceedings [Hoogs et al., 1997].

In November 1996, the RTS was first integrated with the Model-Supported Positioning system, which provides automatic image-to-site model registration. This combination resulted in a successful demonstration of automated, end-to-end imagery exploitation processing; without human intervention, a previously unseen image was registered to a site, IU exploitation algorithms were executed on the image, and feature-specific determinations of change were presented to the IA in a graphic overlay for verification.

The RTS is currently installed at the National Imagery and Mapping Agency (NIMA) in the former National Exploitation Laboratory (NEL). Through the first quarter of 1997, the integrated IU algorithms for site model construction and automated exploitation are being evaluated by imagery analysts (IAs) from NIMA and other government agencies. Although RADIUS did not have sufficient resources for in-depth evaluations of IU algorithms, NIMA is emphasizing algorithm evaluation as a near-term concern. It is hoped that the RTS, with its IU integration framework, will serve as an effective platform for easily presenting new and existing IU algorithms to IAs for evaluation.

Demonstrations, support and moderate extensions of the RTS will continue throughout 1997, and possibly beyond. The RTS contains a large amount of functionality that has yet to be evaluated or assessed by IAs, but this may occur in the coming months. Frequent demonstrations of the system at NIMA are given to government personnel from a variety of organizations.

The RADIUS program has spawned several applications of MSE technology that use the RADIUS software in a more rigorous setting. The SMS adapts IU exploitation to the tactical intelligence problem, while SIAS streamlines the process of generating annotated image products. Both of these technologies are described in the following sections.

3 SMS

The SAIP Advanced Concept Technology Demonstration (ACTD) is a major effort to combine a number of the current DARPA automatic target recognition (ATR) programs into a single architecture. The SAIP imagery exploitation tools are designed to be integrated eventually into or be interoperable with tactical, theater, and national imagery exploitation systems. It includes three imagery exploitation functions: wide area search for specific targets and formations of ground forces, identification and characterization of target vehicles such as tactical ballistic missile launchers, and monitor-

ing of activity at fixed sites. An operational evaluation of the SAIP system is planned for November 1997.

The Site Monitoring System (SMS) is being developed as the site monitoring component of SAIP. It is derived from the RTS, updated to perform exploitation in the tactical environment. Adaptations include improved processing and I/O performance, integration of Synthetic Aperture Radar (SAR)-based ATR algorithms, a user interface for the tactical imagery analyst (IA), integration of automatic image positioning/registration capabilities, and implementation of data and control interfaces to other SAIP elements.

Site monitoring is defined as the activity of repeatedly observing, analyzing, and reporting on significant activity in fixed areas. MSE is well-suited to this task, since it identifies specific geographic regions in an image and executes algorithms on the pixel data based on pre-assigned algorithm profiles [Bremner et al., 1996].

The SAIP program is intended to exploit tactical imagery from Unmanned Aerial Vehicles (UAVs) such as the Global Hawk and Dark Star systems. Due to delays in fielding these UAVs, however, SAIP evaluations are currently using data from the U2 platform, namely ASARS-2 (SAR) and SYERS (EO) imagery. The majority of the SAIP systems will exploit SAR exclusively, but SMS will be monitoring targets first in EO, and later in SAR.

Operations The MSE paradigm presupposes that site models are prepared for the areas of interest before a mission is begun. The analyst is responsible for defining the monitoring profiles and assigning mission priorities, which are used to determine the order of IU execution. Once a mission starts, SMS automates the end-to-end exploitation pipeline that begins after images have been formed by the sensor processing system. Before an IA sees an image, it has been

- geolocated by the sensor
- associated to a site by its geolocation
- registered to a site using photoidentifiable control features
- processed by IU algorithms. For each feature of interest appearing in the image, the feature is:
  - Checked for unacceptable imaging conditions
  - Processed by one or more IU algorithms
- given a priority for viewing based on IU results.

Images are processed according to their preassigned (mission) priority, and the the analyst is presented with images for review based upon a combination of priority and IU processing results. This image prioritization results in optimal use of the analyst's time, since the most important images are placed at the top of the exploitation queue.

Architecture The SMS architecture is shown in Figure 1. The majority of the system is derived from the RADIUS Testbed System (RTS) [Hoogs et al., 1997], ported to a multiprocessing Silicon Graphics Inc. (SGI) Unix platform. SMS performance is improved over the RTS primarily by taking advantage of the SGI's speed and multiprocessing capabilities. SMS is partitioned into two run-time processes, one for executing the IU algorithms and one for supporting the user interface. This division of labor allows the analyst to manually exploit the imagery and report results without affecting the IU processing pipeline.

SMS inherits the main part of its architecture from the RTS. Modifications are being made to harden the code as it moves from a testbed to a demonstration system. As in the RTS, images, site models, IU profiles, and related exploitation data are stored in the site model database [Hoogs and Kniffin, 1994, Kniffin and Hoogs, 1996].

ATR capabilities will be added to SMS by integrating software derived from the the

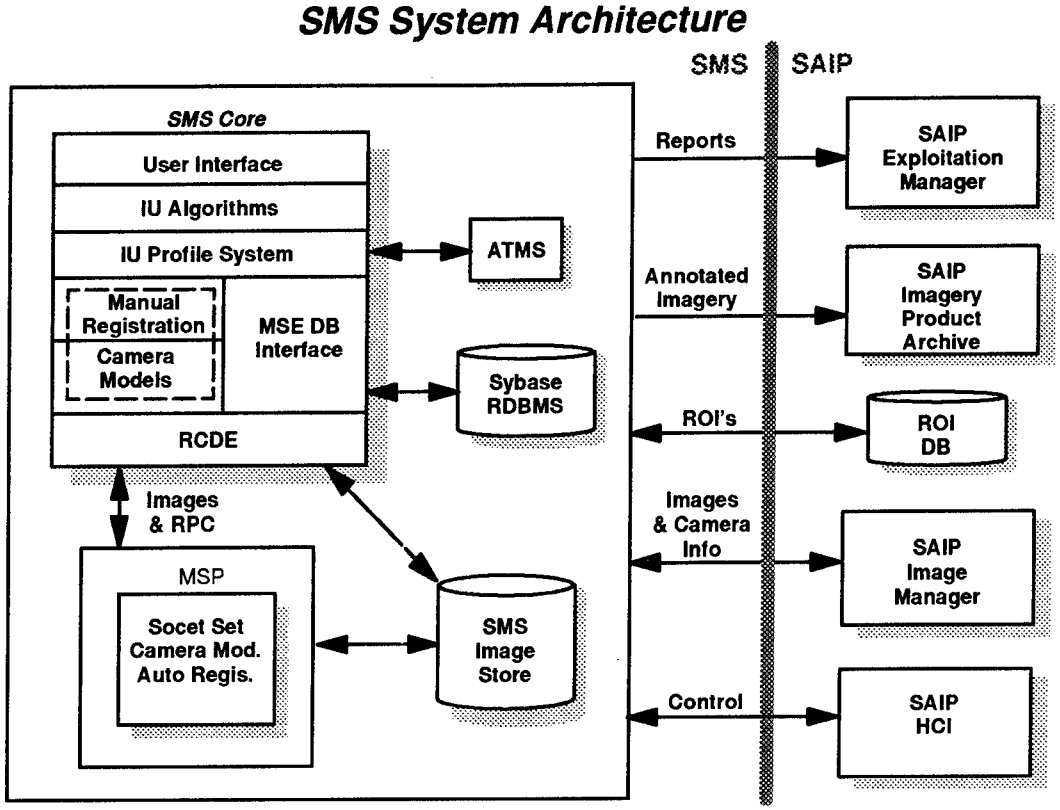


Figure 1: The SMS System Architecture.

Model-Supported Automatic Target Recognition (MOSTAR) and Bosnian Cantonment Area Monitoring System (BCAMS) programs. The ATR system within BCAMS is derived from the Automated Target Monitoring System (ATMS) program. Together, these ATRs should enable SMS to find and identify vehicle targets in SAR imagery.

For SMS to work effectively, each incoming image must be registered to the previously-constructed site model. SMS will use two components in series to perform registration, the SAIP Image Registration component (Harris) and the Model-Supported Positioning system (GDE/TI).

SMS is primarily a stand-alone system with a few critical interfaces to SAIP. Components to the right of the shaded line in Figure 1, such as imagery inputs and reporting outputs, are external to SMS. For example, the region-of-interest (ROI) database supplies geolocations of targets found by the main SAIP system, to allow analyst to correlate site monitoring and wide-area search results.

User Interface The SAIP user interface allocates two displays to each analyst, so SMS has chosen to allocate one screen to an overview function and one screen for detailed image exploitation. The overview screen, illustrated in Figure 2, is dominated by an overview pane for displaying an image of the site of interest with graphic overlays of the geographic regions being monitored. The panel notifies the analyst when a new image has been received and processed by the IU subsystem, and enables the new image to be examined. Each monitored region is colored according to the results found by the IU processing. For example, results of large change, high confidence, and high priority are outlined

in red. At the same time, textual descriptions of the IU results on that image are displayed at the bottom of the panel. By showing an overview of the entire site with color-coded overlays, the analyst is immediately alerted to the pattern of activity at the site.

As each processed region is selected with the cursor, the image to be exploited is zoomed to fit the region of interest on the exploitation screen as illustrated in Figure 3. The analyst can then visually exploit the image to confirm (or deny) the accuracy of the IU results, using standard electronic light table tools to manipulate the image display. Further information regarding the profile results for that particular region is displayed at the bottom of the panel. This screen allows the analyst to display graphic results of the IU processing overlaid on the feature, such as crosshairs outlining detected vehicles. It is also the jump-off point for more complex analyses of IU results such the display of trends over time.

After reviewing the imagery, the analyst is required to report on the observed activity. SMS will use the results of IU processing and the user-defined profile to automate part of the process of text reporting. Graphic overlays, including both the site models and the IU results, can also be combined with the image data for generating annotated imagery products. The goal of SMS is to make the task of exploiting imagery and reporting on the results as efficient as possible.

IU Community Benefits By participating in the SAIP ACTD, SMS will move the technology integrated under the RADIUS program one step down the path to operational use. Similar to RADIUS, SMS is not supplied by one contractor. Rather, it is a combination of IU technologies from the wide range of institutions participating in RADIUS, from which algorithms are being selected according to their tactical utility.

The RADIUS software can automatically execute a large number of IU profiles on imagery that is already part of a site model, but the front-end image ingest and registration process was not fully completed. Combined with SAIP, SMS will complete the end-to-end automated exploitation process by interfacing to image sources and automatic registration, forming an excellent conduit for evaluating IU algorithms on a large volume of imagery. A one-day operational mission will collect tens of gigabytes of data to be registered, filtered, and have IU algorithms executed on it.

Because the SMS code is derived from RADIUS, most components will be integrated back into the RTS so that the research testbed evolves and improves. By maintaining the RADIUS exploitation framework applications programmer interface (API), software can be easily transitioned from the research testbed to the SMS demonstration system. SMS therefore represents an ongoing opportunity for IU community involvement; as on RADIUS, IU technologies are solicited for their inclusion in SMS for evaluation on operational data. Continued support from the organizations responsible for the success of RADIUS will help ensure the success of SMS.

SMS development is funded by DARPA, and includes team members SRI International and General Electric Corporate Research and Development.

4 SIAS

An example of technology transfer from the RADIUS program, the Spatial Image Annotation System (SIAS) is a short-term effort to deploy an operational-prototype, MSE workstation based on RADIUS technologies. SIAS introduces MSE to real-world exploitation environments. and will be the first time that RADIUS-developed technology is used operationally. SIAS incorporates the fundamental spatial framework of RADIUS to provide three-dimensional modeling capabilities and the capability to project 3D models onto images. Although SIAS does not include any IU algorithms from RADIUS, it does provide an operational

Figure 2: Prototype SMS Overview Screen

Figure 3: Prototype SMS Exploitation Screen

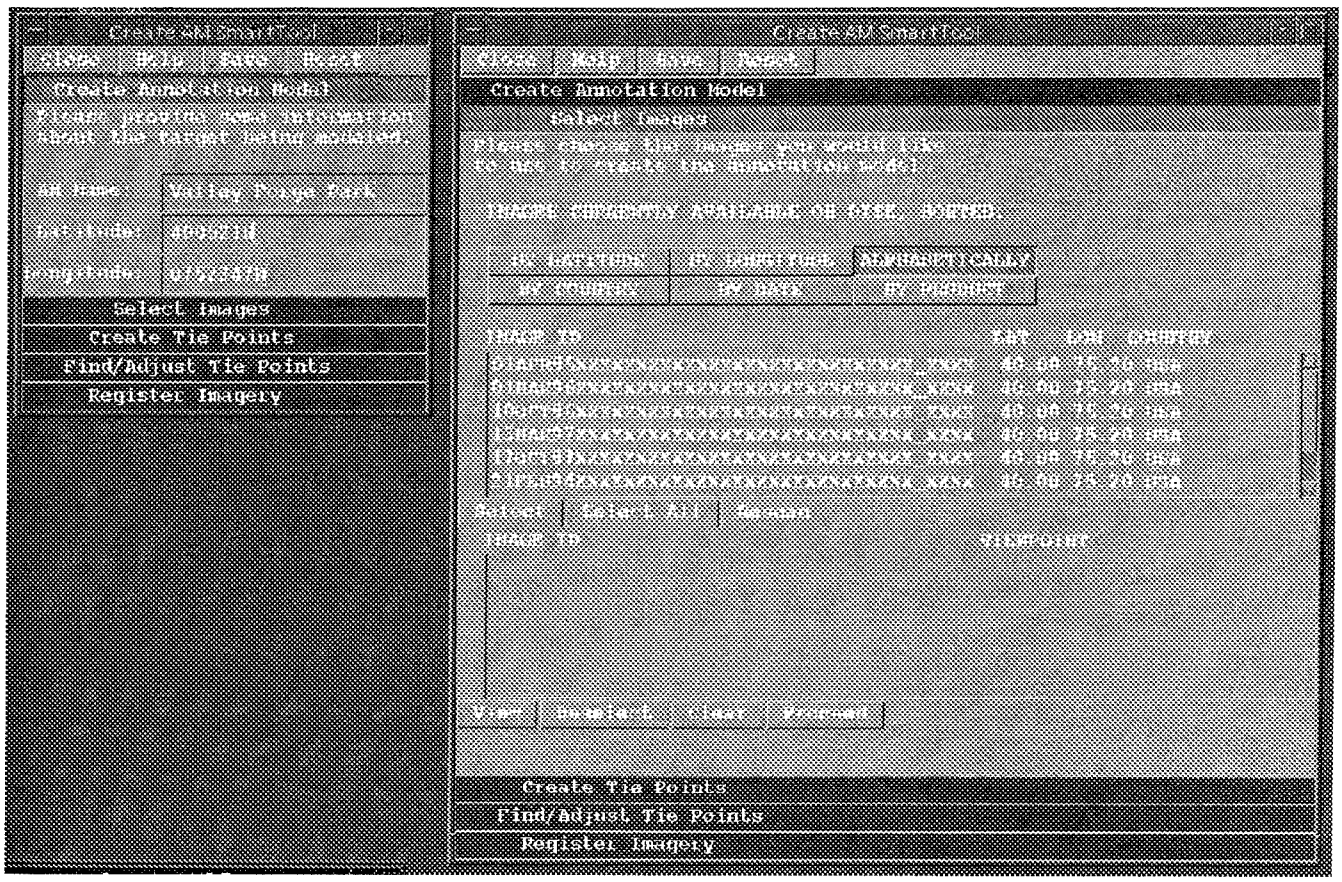


Figure 4: The SIAS SmartTool.

introduction to fundamental IU concepts related to the RADIUS concept of site models.

SIAS was selected for development by an intelligence community effort to transition promising technology from R&D to usable tools for analysts. In both 1995 and 1996, analysts participating in these evaluations quickly grasped the importance of MSE, and SIAS was rated very highly. Analysts continued to play a key role in the development of SIAS, particularly the user interface components. This was critical to the successful transition from R&D testbed to operational system.

One of the key areas which needed to be addressed was the process of creating a three-dimensional world, which is the basis for MSE. In RADIUS, this process involved a large amount of effort on the part of IU scientists to achieve the desired accuracy. SIAS uses a two-part approach to address this issue: first, by not including IU algorithms, we can relax somewhat the requirements for precise registration. The idea is to use SIAS to outline general areas of intelligence interest in an image, and not individual structures. To reflect this difference, in SIAS we use the term *Annotation Model;* this is intended to distinguish it from the more rigorous, detailed, and extensible RADIUS site model. The second part of the SIAS approach to building an annotation model (AM) was to provide a sequenced, step-by-step approach to AM construction, and to incorporate this framework into a tool which can be used by the analyst.

Figure 4 shows the result of this process. Based on help facilities used in PC applications, this is called the "SmartTool." The SmartTool guides the user through the various steps of creating the AM, keeps track of the analyst's progress through the procedure, and also offers help

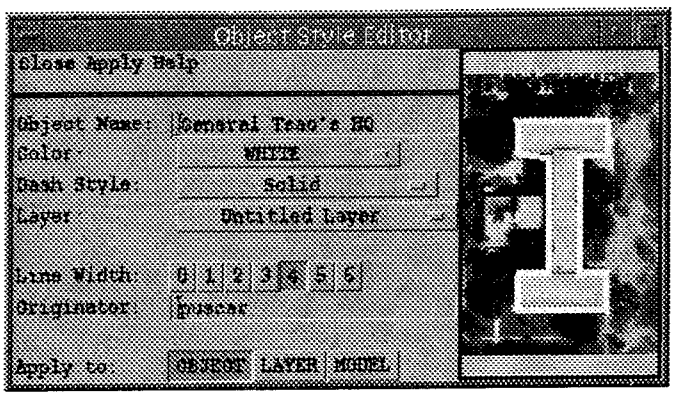


Figure 5: The SIAS Style Manager.

through pop-up messages and an on-line hypertext user's manual.

Another emphasis of SIAS which differed slightly from RADIUS is the need to create annotated images for briefings, reference, or collaborative analysis. RADIUS provides a wide range of capabilities for systematically annotating the elements of a site model; for SIAS, the most commonly used of these were incorporated into a Style Manager (Figure 5), which provides a single interface to manipulating the appearance of objects. Also note that, while RADIUS provides many dozens of different object types, SIAS provides only a few basic entities.

As noted above, potential users of SIAS have been involved in the design process though design reviews and hands-on evaluations. Although SIAS does not include automated site-model construction tools, analysts did see the value of these and indicated they would like to see them in future versions. Similarly, analysts also saw the value of time-based profiles, since they also requested the snippet sequence tool developed for RADIUS. What is exciting about these comments is that the analysts generated these ideas without any knowledge of the full RADIUS capabilities: they were able to understand the goals of IU related to MSE, and to develop new concepts based on the introduction they saw in SIAS. This is promising for the expansion of IU technologies in future applications.

SIAS does point out some areas which need further development. Automated site-model construction tools need to be made more robust and general before they can be used operationally. The image registration problem also needs to be refined, although the Model-Supported Positioning project being incorporated into RADIUS holds promise in this area. Analysts make extensive use of database systems; these can be linked to the MSE system, and even integrated into the IU framework. Finally, in order to make 3D model-derived information more widely useful, an interoperable standard needs to be developed which will allow users to build an Annotation Model or Site Model, and then export the data to other applications.

SIAS will be delivered to three government sites in Spring 1997. It will be evaluated and used for at least 3 months, and may be enhanced with a second stage of development based on analyst feedback. SRI International is our subcontractor, providing support and development on the RADIUS Common Development Environment.

5 IR&D

In addition to contract activities, we are engaged in independent research and development (IR&D) that complements contract technology growth by developing new capabilities applicable to a range of programs. Our IR&D efforts have focused on IU research, including frameworks for the management of IU subsystems, and prototype development of persistent storage for model-supported systems.

5.1 IU Research

Our IU research is focused on developing new algorithms that fully exploit site model context, particularly historical imagery. We have developed change detection algorithms that identify changes in man-made structures, such as buildings, roads, and construction areas. The algorithms also detect the disappearance or absence of modeled features, and hence can be used to monitor vehicles parked in specific locations, such as aircraft on a tarmac. The system is described elsewhere in these proceedings [Hoogs, 1997], and is summarized here.

The main contribution of our approach is that low false alarm rates are attained by learning appearance characteristics from historical imagery [Hoogs and Bajcsy, 1995]. In outdoor scenes, the appearance of structures varies considerably based on weather, season, and imaging parameters such as viewpoint and illumination. Structures can be detailed with surface features such as windows, doors, vents, and albedo changes. Typically, geometric models of structures do not capture these smaller features or photometric features. Hence a change detection system that relies on geometry alone may be prone to systematic error. For example, a building that has similar albedoes on its walls and roof (and even the background) may consistently give rise to change indications, because its roof-lines are not detected in the image. Or, a building with lots of unmodeled superstructure may result in disjointed segmentations, leading to false alarms.

The goal of our system is to identify true changes in structures while ruling out apparent differences due to non-geometric effects. We have observed that photometric features are often consistent enough, within certain constraints, to be locally predicted and accounted for. Training imagery is used to establish probabilistic models of the appearance of geometrically modeled features. Within each aspect of a model edge, appearance characteristics along the edge are characterized, providing an implicit modeling of the appearance of complex surface

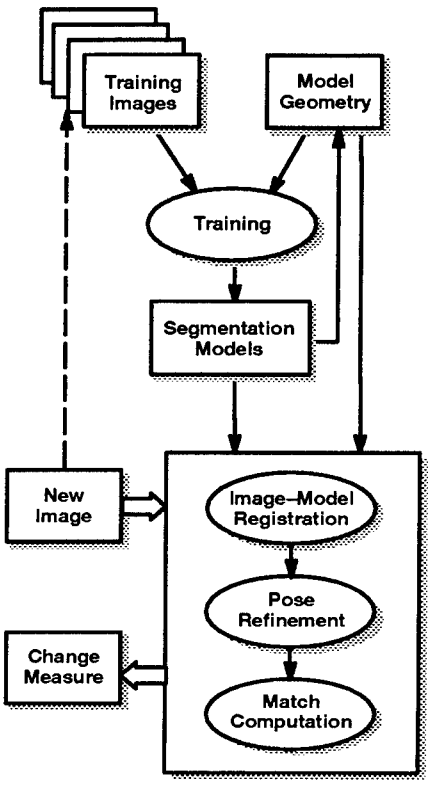


Figure 6: Change detection process flow.

features. In new imagery, these models are used to account for missing edges, albedo changes, and other segmentation-level phenomena.

In many cases, few training images are required. By using both model geometry and training images, we allow incremental improvement in accuracy as more images are examined (and verified). The system works with no training imagery, using only the geometric model, or any number of training images with any imaging conditions.

The change detection system is shown in Figure 6. Initially, segmentation models are created using the model geometry of structures in a scene and registered training images of the scene. When a new image is presented to the system, it is registered to the geometric models or other scene features using manual or automatic techniques.

Many existing image registration algorithms, manual or otherwise, typically result in at least 2 pixels of error because of unmodeled sensor

distortions and noise. To compensate for this, the system performs a local 2D translational pose refinement step that adjusts the position of a single object model with respect to the image [Hoogs and Bajcsy, 1996, Hoogs, 1996].

The next stage of the algorithm, match computation, is a virtual operation; the pose refinement stage actually computes the match score that is the output change measure. The same matching algorithm or metric between image and object model is used in pose refinement and change detection, so that the optimal value found in pose refinement is considered to be the best estimate for the change measure. The match metric is computed using the geometric model and the learned appearance characteristics.

The system has been tested on many images of multiple sites. The results indicate that the system performs well at mitigating false alarms in change detection without reducing the probability of detecting change, in comparison to purely geometric methods. The contribution of training imagery is particularly evident when the system is applied to visually complex features, such as buildings obscured by adjacent trees.

5.2 Persistent Support for MSE

Over the past three years we have invested considerable effort in providing persistent support for model-supported exploitation and associated IU algorithms [Kniffin and Hoogs, 1996, Hoogs and Kniffin, 1994]. More recently, our work has focused on performance issues, historical queries and advanced support for IU systems. For further discussion, see [Cardenas, 1996].

References

[Bremner et al., 1996] B. Bremner, A. Hoogs and J. Mundy. Integration of Image Understanding Exploitation Algorithms in the RADIUS Testbed. *Proceedings of the ARPA IU Workshop*, Feb. 1996.

[Cardenas, 1996] R. Cardenas and A. Hoogs. The RADIUS Testbed Database: Temporal Queries and Optimization. *Proceedings of the SPIE Conference on Applied Imagery and Pattern Recognition: Emerging Applications of Computer Vision*, pp. 192-200, 1997.

[Hoogs, 1997] A. Hoogs. Combining Geometric and Appearance Models for Change Detection. *These Proceedings.*

[Hoogs et al., 1997] A. Hoogs, W. Bremner and D. Hackett. The RADIUS Phase II Program. *These Proceedings.*

[Hoogs, 1996] A. Hoogs. Pose Adjustment Using a Parameter Hierarchy. *Proceedings of the ARPA IU Workshop*, Feb. 1996.

[Hoogs and Bajcsy, 1996] A. Hoogs and R. Bajcsy. Model-Based Learning of Segmentations. *Proceedings of the International Conference on Pattern Recognition*, Vienna, Austria, 1996.

[Hoogs and Bajcsy, 1995] A. Hoogs and R. Bajcsy. Using Scene Context to Model Segmentations. *Proceedings of the IEEE Workshop on Context-Based Vision*, Cambridge, MA, 1995.

[Hoogs and Kniffin, 1994] A. Hoogs and B. Kniffin. The RADIUS Testbed Database: Issues and Design. *Proceedings of the ARPA IU Workshop*, Nov. 1994.

[Kniffin and Hoogs, 1996] B. Kniffin. and A. Hoogs. Database Support for Exploitation Image Understanding. *Proceedings of the ARPA IU Workshop*, Feb. 1996.

IU AT UI: AN OVERVIEW OF RESEARCH DURING 1996-97

Narendra Ahuja and Thomas Huang
Beckman Institute and Coordinated Science Laboratory
University of Illinois
405 North Mathews Avenue
Urbana, Illinois 61801

Abstract

This paper presents an overview of the research in image understanding (IU) at the University of Illinois (UI) conducted during 1996-97. During this period, our work has been in five areas: integration, segmentation, image compression and resolution enhancement, motion analysis, and representation and recognition. Work in each of these areas is reviewed.

1 Introduction

A major part of our research since [6] is in five areas (Secs. 2-6). The first area (Sec. 2) is concerned with integration of multiple image cues in performing image interpretation. These cues capture different aspects of the three-dimensional (3D) scene structure, and their integrated analysis leads to a more robust inference about the scene characteristics than possible from individual cues. The second area (Sec. 3) addresses the problem of low level image segmentation. The emphasis here is on obtaining automatic, hierarchical descriptions of the low level structure. We have used the detected, multiscale image structure for image compression, and enhancement of image resolution; this constitutes our work in the third area (Sec. 4). The fourth area (Sec. 5) is about our continuing work on interpretation of image sequences showing dynamic scenes. Here we consider the problems of detecting feature correspondences and estimating the 3D motion parameters and surface structure from correspondences in a sequence of images showing rigid as well as nonrigid motion. The fifth area (Sec.6) describes our recent work in image representation and recognition. Representative projects in each of these areas are summarized in the following sections.

This research was supported in part by the Advanced Research Projects Agency under grant N00014-93-1-1167 administered by the Office of Naval Research and the National Science Foundation under grant IRI-93-19038.

To keep the paper brief, we have not included discussions of, and references to, relevant work done by others. Such discussion and references are available in the listed publications.

2 Integration

Our goal in this area is to perform image interpretation such that the interpretation simultaneously satisfies a range of constraints imposed by the image structure and the model of the scene. To do this, we use different computational processes each of which carries complementary or redundant information derived from different image cues. Image interpretation is the result of a cooperative computation that resolves conflicts and ambiguities arising from the individual processes. We have presented several examples of the integration approach in previous IU workshops [1, 2, 3, 4, 5, 6]. Here we summarize some recent work on integration.

2.1 Integrated Active Stereo

Our previous work has been concerned with fixation of different objects in the scene, and surface estimation for each object from multiple stereo cues such as focus, vergence and disparity [16]. Recently, we have investigated the problem of efficient fixation. Given a target, fixation of an active camera pair requires that the pan and tilt angles must be set to bring the target to image centers. However, calibration of real cameras involves tedious estimation of a number of imaging parameters. Fortunately, calibration is not essential for fixation if images are acquired and used as feedback during the fixation process to continuosly direct the cameras to the target. We have used a direct mapping from the changes in camera (or joint) angle space to the direction of the resulting target motion in the image plane, to determine changes in camera angles necessary to reduce the image plane disparity between image center and the target location. In addition to the calibration parameters, the use of the mapping also incorporates other unmodelled effects such as deviations from the

assumed imaging model. The mapping is formulated as a task in nonlinear function approximation, and, for computational efficiency, learnt from real data at multiple resolutions; the coarse levels are concerned with large changes and the fine levels with small changes. Fixation is accomplished by first executing large angle changes and slowly reducing their magnitudes to converge on the final camera orientations. The sensing and use of the feedback are made in two modes: continuous and intermittent. Learning is performed using a neural network. Details can be found in [32].

2.2 Nonfrontal Imaging Camera

We have continued our work on nonfrontal imaging. An omnifocusing nonfrontal imaging camera (NICAM) can provide an in-focus image of an arbitrarily wide scene with all objects appearing in focus regardless of their locations in the scene (Fig. 1). Further, a range (from focus) estimate of each visible scene point is also derived [13]. The camera's sensor plane is not perpendicular to the optical axis as is standard. This special imaging geometry eliminates sensor plane movement usually necessary for focusing. Camera panning, required for panoramic viewing anyway, in addition enables focusing and range estimation. Thus panning integrates both standard mechanical actions of focusing and panning, implying range estimation at the speed of panning. An advanced prototype of NICAM has been developed (Fig. 2). In [25], we describe strategies for optimal selection of panning angle increments and sensor plane tilt for NICAM. We have also investigated the use of standard cameras for acquiring panoramic images. We have developed methods to optimize the image acquisition strategy in order to reduce redundancy. We show that panning a camera about a point f (focal length) in front of the camera eliminates redundancy. [25] shows some panoramic images acquired using the standard camera.

2.3 Integrated Stereo and Shading

We have continued our work on joint surface estimation from the complementary cues of stereo and shading. We have developed a method in which an empirically determined associative model relating appearance to surface shape is used [24]. The parameters are estimated using examples provided by a stereo algorithm. Through a scale change, the statistically estimated model is made to be more accurate than the algorithm that generates the examples. The method is a generalization of shape from shading methods that does not rely upon idealized models of the image formation process. It more accurately recovers small surface detail than is possible with methods such as stereo and motion. The function estimation is done by methods that are similar to the methods of pattern recognition. Indeed, this approach is the continuous analogue of pattern recognition and is closely related to methods of joint space learning used in robotics.

2.4 Integrated Region and Border Detection

We have developed the transform for multiscale detection of image structure, which we introduced in earlier work [6]. The transform extracts image regions at all geometric and photometric scales. Linear approaches such as convolution and matching have the fundamental shortcoming that they require *a priori* models of edge geometry. The transform we have proposed avoids this limitation by letting the structure emerge, bottom-up, from interactions among pixels, in analogy with statistical mechanics and particle physics. The transform involves global computations on pairs of pixels followed by vector integration of the results, rather than scalar and local linear processing. An attraction force field is computed over the image in which pixels belonging to the same region are mutually attracted and the region is characterized by a convergent flow. The transform possesses properties that allow multiscale segmentation, or extraction of original, unblurred structure at all different geometric and photometric scales present in the image. This is in contrast with much of the previous work, wherein multiscale structure is viewed as the smoothed structure in a multiscale decimation of image signal. Scale is an integral parameter of the force computation, and the number and values of scale parameters associated with the image can be estimated automatically. Regions are detected at all, *a priori* unknown, scales resulting in automatic construction of a segmentation tree, in which each pixel is annotated with descriptions of all the regions it belongs to. The transform provides a general approach to multiscale, integrated edge and region detection, or low-level image segmentation.

The basic operation of the transform is to convert the image I into a vector field $\mathbf{F}$. The vector $\mathbf{F_p}$ at an image location $\mathbf{p}$ is defined as

$$\mathbf{F_p} = \int_{\mathbf{q} \neq \mathbf{p}} d_s(\mathbf{r_{pq}}, \sigma_s(\mathbf{p})) d_g(\Delta I, \sigma_g(\mathbf{p})) \hat{\mathbf{r}}_{\mathbf{pq}} \mathbf{dq} \quad (1)$$

where

$\hat{\mathbf{r}}_{\mathbf{pq}}$ = unit vector in the direction from $\mathbf{p}$ to another image location $\mathbf{q}$;

$\sigma_s(\mathbf{p})$ = spatial scale parameter at $\mathbf{p}$; related to the shortest distance to region boundary; all valid $\sigma_s(\mathbf{p})$ values are computed automatically;

$\sigma_g(\mathbf{p})$ = photometric scale parameter at $\mathbf{p}$; denotes contrast of region with surround; all valid

$\sigma_g(\mathbf{p})$ values are computed automatically;

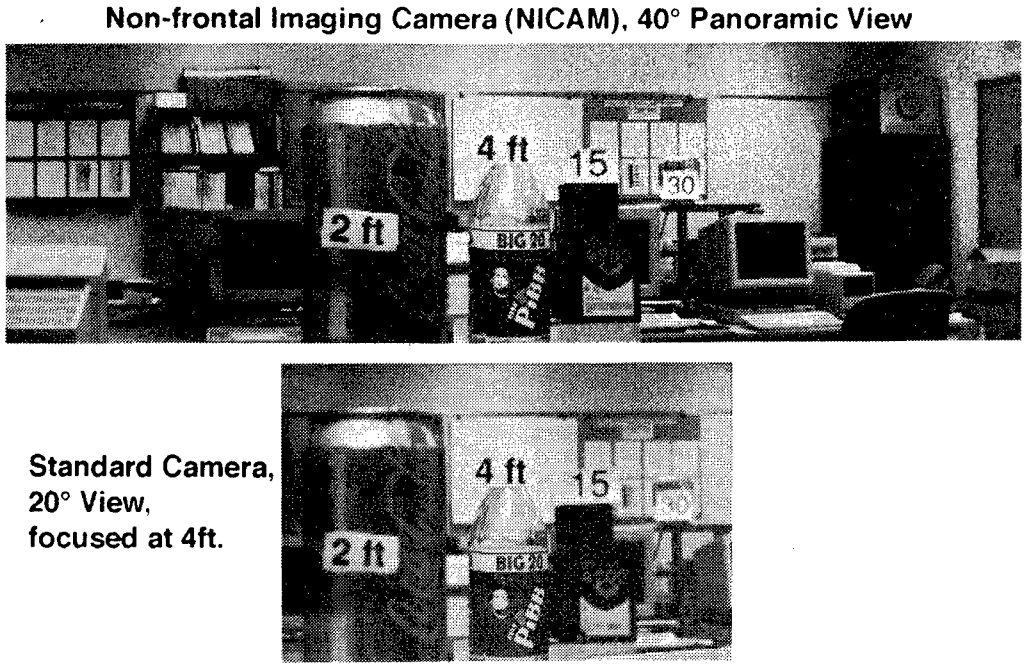


Figure 1. Comparison of the performance of NICAM with a regular camera. The upper row shows a 40-degree image acquired by NICAM and the lower row shows a 20-degree view of the same scene acquired using a regular camera having a visual field of 20 degrees and focused at 4'. While all the objects in the upper row appear focused, the objects at distances other than 4' in the lower row are defocused to different degrees, depending on their distances relative to the focus distance of 4'.

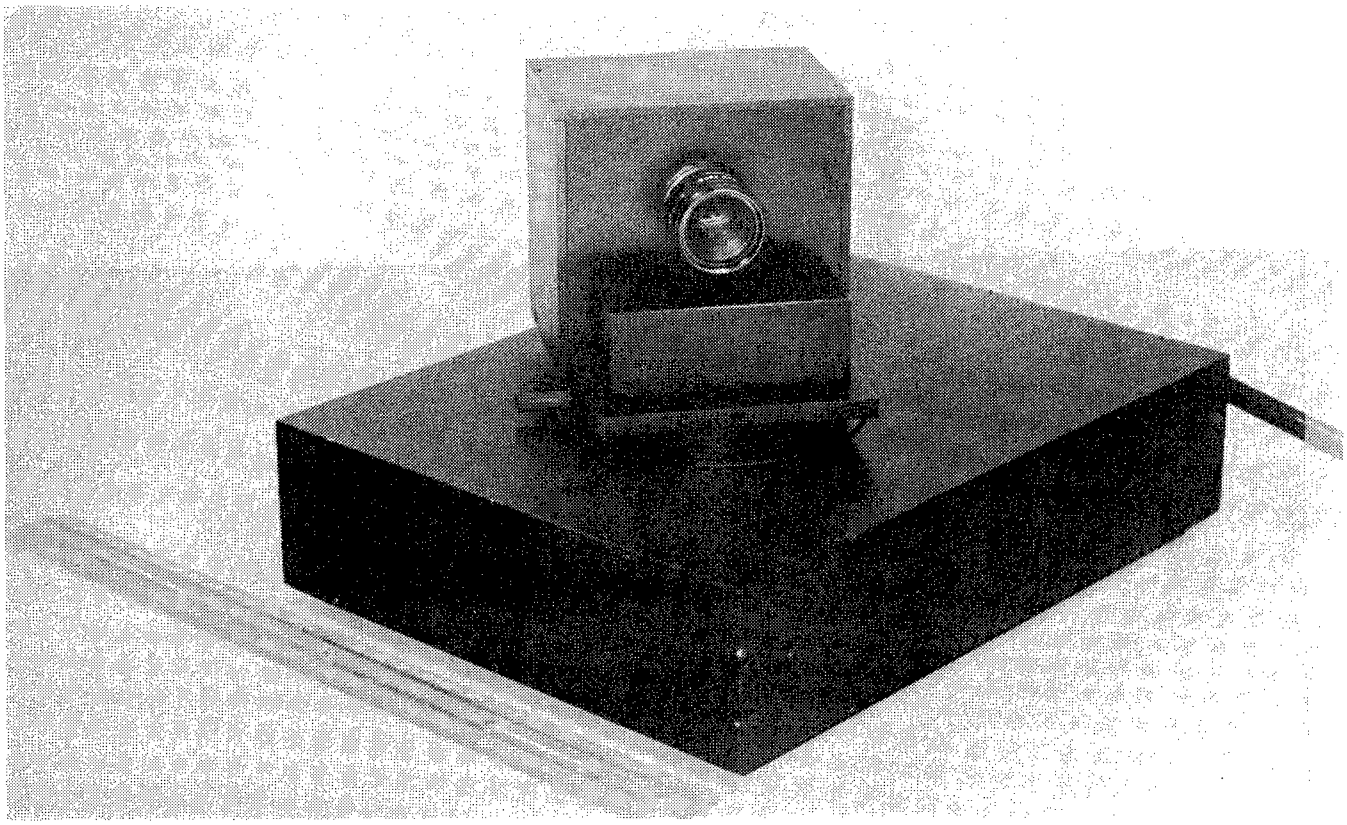


Figure 2. The current prototype NICAM.

ΔI = absolute gray level difference between image points under consideration;
$d_s(\mathbf{a}, b)$ = A nonnegative, nonincreasing function of $||\mathbf{a}||$, not identically 0 for $||\mathbf{a}|| \leq b$, and 0 for
$||\mathbf{a}|| > b$, and
$d_g(a, b)$ = A nonnegative, nonincreasing and symmetric function of a, not identically 0 for $a \leq b$,
and 0 for $a > b$.
Since $d_g(a, b)$ as defined above cannot be a linear function of a for unrestricted values of a, the transform does not obey the principle of superposition and hence is nonlinear. Details can be found in [19, 14].

3 Segmentation

We have used the transform summarized above for segmentation of multidimensional, multivariate images. [18] describes an algorithm for 2D image segmentation at multiple scales. The detected regions are homogeneous and surrounded by closed edge contours. Other approaches to multiscale segmentation have represented an image at different scales using a scale-space. However, structure is only represented implicitly in these approaches, structures at coarser scales are inherently smoothed, and the problem of structure extraction is unaddressed. We argue that the issues of scale selection and structure detection cannot be treated separately. A new concept of scale is presented which represents image structures at different scales, and not the image itself. This scale is integrated into the transform which makes structure explicit in the transformed domain. Structures which are stable (locally invariant) to changes in scale are identified as being perceptually relevant. The transform can be viewed as collecting spatially distributed evidence for edges and regions, and making it available at contour locations, thereby facilitating integrated detection of edges and regions without restrictive models of geometry or homogeneity. In this sense, it performs Gestalt analysis. All scale parameters to the transform are automatically determined, and structure of any arbitrary geometry can be identified without any smoothing, even at coarse scales.

We have used the transform to segment 3D volume data obtained by magnetic resonance imaging [28]. This preliminary work presents a new method for multiscale segmentation of volume images. The segmentation leads to well-characterized 3D regions at different spatial and intensity scales. The detected regions are closed and are homogeneous relative to their surround. A tree is generated containing the region information extracted across a range of homogeneity scales and represents the multiscale volumetric structure.

To compare the transform performance with alternate algorithms which group points/pixels, we have investigated a new framework for hierarchical segmentation of multivariate multidimensional functions into homogeneous regions [31]. Homogeneity is defined as constancy of n-th order derivatives (called features) of the function. A regular multidimensional grid of sample points in the domain of the function is partitioned based on similarities of region features at the sample points. Three other segmentation techniques and applications to one-, three- and six-variate data in two- and three-dimensions are described for the zeroth and first order region features. Details can be found in [39]. [26] address the special case of irregularly sampled data and develop a clustering algorithm for dot patterns in n-dimensional space. The n-dimensional space is viewed as representing a multivariate (n_f-dimensional) function in a n_s-dimensional space ($n_s + n_f = n$). The algorithm decomposes the clustering problem into the two lower dimensional problems. Clustering in n_f-dimensional space is performed to detect the sets of dots in n-dimensional space having similar n_f-variate function values (location based clustering using a homogeneity model). Clustering in n_s-dimensional space is performed to detect the sets of dots in n-dimensional space having similar interneighbor distances (density based clustering with a uniformity model). Clusters in the n-dimensional space are obtained by combining the results in the two subspaces. Extensions of the approach to the case of texture is presented in [35] which describes a new hierarchical texture segmentation method based on (1) finite sets of first-order statistics (texture attribute dictionaries) and (2) second-order statistics (spatial co-occurences) calculated from global, spatially irregular regions found in an anisotropic fashion. A textured region is viewed as a set of uniformly distributed primitives, whose attribute homogeneity model provides a framework for the multiscale analysis. Robustness against (1) noise in primitives, (2) attribute dictionary overlaps, (3) nonuniform distribution of primitives and (4) variable size of attribute dictionaries, and computational efficiency of the method are presented. Finally, [33] describe a general method for hierarchical segmentation of multivariate multidimensional functions. The method partitions a regular multidimensional grid of sample points in the domain of a multivariate function based on similarities of multivariate function values (attributes) at the sample points. Similarity of two attributes is modeled by their Euclidean distance and is called as the homogeneity of the attributes. A connected set of sample points from the grid is said to define a region in the final partition (segmentation) if (1) attributes of all interior samples are no more than a given value δ apart, and (2) attributes of all samples outside the region are no closer than a given value

α. The problem of segmenting in n-dimensional can be decomposed into lower-dimensional segmentation problems which makes the method computationally efficient. Segmentation results are represented in the form of a tree formed by regions detected for various values δ. Experiments have been conducted that demonstrate the noise robustness and computational efficiency of the segmentation, and compare its performance with three other segmentation techniques. The method has been applied to 2D and 3D medical data, botanical data and satellite data.

In another approach to segmentation, we have extracted regions as a union of a priori chosen primitive shapes. We use ellipses of relatively uniform gray level as shape primitives. We use scale-space concepts to fit ellipses over a range of sizes, eccentricities, and orientations. A region extractor is presented in [27] which uses filters based on the elliptical Gaussian to find homogeneous elliptically-shaped regions in real images. A filter which is similar to the Laplacian of the elliptical Gaussian is applied to the image to locate possible ellipse centers. A scale-space technique is used to verify that these detected sites are true ellipse centers. Two other filters, related to the first by differentiation with respect to the scale parameter, are applied to these potential ellipse sites to eliminate regions which are not sufficiently elliptical. With these three filter responses, the size and contrast of the region are computed.

[41] presents a new 2D edge detection algorithm. The algorithm detects edges in 2D images by a curve segment based edge detection functional that uses the zero crossing contours of the Laplacian of Gaussian (LOG) as initial conditions to approach the true edge locations. We prove that the proposed edge detection functional is optimal in terms of signal-to-noise ratio and edge localization accuracy for detecting general 2D edges. In addition, the detected edge candidates preserve the nice scaling behavior that is held uniquely by the LOG zero crossing contours in scale space. [46] presents a new color image edge detection algorithm. By exploiting the statistical properties of a given image, global information of the image is extracted to guide the local gradient computation. Cluster analysis is first performed in the 3D color space to find the major chromatic components of the image. According to these clusters, groups of linear chromatic transforms are generated. The edges are treated as the transitions from one cluster to another. To maximize the gradient magnitude, an appropriate chromatic transform is chosen for each pixel.

4 Image Compression and Resolution Enhancement

We have used the detected image structure for two image-space applications described below.

4.1 Compression

Different methods have been proposed to achieve lossless compression, the more successful of which exploit the *local* two dimensional redundancies in the images. Most methods however do not produce significantly and consistently better results than the simple JPEG implementation. This can be attributed to the overhead generated by such methods negating any advantage accrued from obtaining a better residual. We have used the high fidelity of our transform-based segmentation for lossless compression. [37] uses segmentation information in order to form a smoothly varying residual image, which is devoid of edges, and then uses an autocorrelation model to further decorrelate the residual. Each region generated by segmentation is further subdivided into the interior sub-region and the edge sub-region; the distribution of grey level values in the sub-regions being distinctly different. The pels in the edge sub-regions, which have a higher standard deviation (with respect to the interior sub-regions), can be modeled explicitly using edge models. This is feasible because the 2-D variation of the grey level values within the edge sub-regions follows a specific distribution. The interior sub-region for each region are, as a first approximation, modeled by a constant. Specifying the models for the interior and edge sub-regions of each region models the entire image. Subtracting this model from the original image we obtain a smoothly varying residual, which is then modeled by autocorrelation based minimum variance prediction.

4.2 Resolution Enhancement

We have used the extracted structural description of an image to enhance its resolution, e.g., to magnify a small image to several times its original size while avoiding blurring, ringing or other artifacts. Classical methods include bilinear, bi-cubic or FIR interpolation schemes followed by sharpening using methods such as unsharp masking. Interpolation schemes tend to blur the images when applied indiscriminately. [37, 40] describe a method we have developed based on the projection on convex sets (POCS) formalism. POCS is used to find a solution which lies at the intersection of various convex constraint sets that restrict the locations of edge and nonedge pixels. In a related other effort, we have addressed the problem of applying a multi-dimensional linear transform over an arbitrarily shaped support. The usual practice is to fill out the support to a hypercube by zero padding. This does not however yield a satis-

factory definition for transforms in two or more dimensions. The problem that we have considered is: how do we redefine the transform over an arbitrary shaped region suited to a given application? We present a novel iterative approach to define any multi-dimensional linear transform over an arbitrary shape given that we know its definition over a hyper-cube. Our proposed solution is (1) extensible to all possible shapes of support (whether connected or unconnected) and (2) adaptable to the needs of a particular application. We also present results for the Fourier Transform, for a specific adaptation of the general definition of the transform which is suitable for compression or segmentation algorithms.

[30] considers another aspect of resolution enhancement: removing image blur. Images are assumed to be obtained from a planar, stationary object in a frontal plane with respect to the camera. The cases of coherent, non-coherent and partially coherent imaging with quasi-monochromatic and polychromatic illumination are considered. We show that blurring is not a linear process for imaging an extended non-coherent object with a camera within the thin lens approximation using polychromatic illumination (all previous implementations of deblurring/depth from deblurring algorithms have considered blurring as a linear process). This follows from the fact that a blurred image cannot be considered as formed from a non-coherent wavefront; the wavefront is partially coherent. It is specifically shown that image blurring is a non-linear process in 1) the general case of Huygen-Fresnel point spread function and 2) the practically applicable case of the geometrical optics approximation. As conclusion, it is found that deblurring in its most general form is not amenable to current theoretical or practical methods.

In [47], we present a new multi-scale image warping method based on the weighted Voronoi diagram. Weights are assigned to the control points according to their influence scales. At each scale level, a triangulation based on the weighted Voronoi diagram is constructed. Then the interpolation of displacements is performed on this triangulation. The advantage of this approach is that the underlying triangulation changes between scales to fit the warping scale. Both global warping and local warpings can be modeled appropriately using this approach.

5 Motion Analysis

5.1 Detecting Feature Correspondences and Matching

The problem of feature correspondences and trajectory finding for a long image sequence has received limited attention in the past. Most attempts involve small numbers of features and make restrictive assumptions such as the visibility of features in all the frames. In our earlier work, a coarse-to-fine algorithm was described to obtain pixel trajectories through the sequence [10]. The algorithm uses a coarse scale point feature detector to form a 3-D dot pattern in the spatio-temporal space. Increasingly dense correspondences are obtained iteratively from the sparse feature trajectories. At the finest level, matching of all pixels is done using intensity correlation and the finest boundaries of the moving objects are obtained. The trajectories are extracted as 3D curves formed by the points using perceptual grouping. The trajectories obtained are then segmented into subsets corresponding to distinctly moving objects [10]. Our previous work on trajectory detection using Hopfield networks is reported in [15].

We have done preliminary investigation of the use of image segmentation derived using the transform reviewed earlier to structure detection in video. We have addressed the simpler case of matching pairs of video frames instead of treating the video as 3D data. There are two stages to this work. First, regions are matched across a pair of frames using a graph matching formulation [36]. Three preselected values of homogeneity scale are used as indexes into the segmentation tree of each image to produce three different image partitions. Each pair of partitions at the same scale are matched from coarse to fine, with coarser scale matches guiding the finer scale matching. Each partition is represented as a region adjacency graph, within which each region is represented as a node and region adjacencies are represented as edges. Region matching at each scale then consists of finding the set of graph transformation operations (edge detection, edge and node matching, and node merging) of least cost that create an isomorphism between the current graph pair. Second, an affine transformation is computed for each set of matched regions, at all scales. The change in shape of the regions is estimated and used in computing a motion field at each scale. This yields a rough estimate of the motion field. As a test of the detected field, we have attempted to recover 3D motion and dense structure of the objects in the image sequence [34]. The algorithm first estimates motion and partial structure of the scene from the affine parameters. This first-order flow-based information is then used to obtain a dense estimate of 3D structure. Finally, shading information is used to refine the estimated dense structure.

5.2 Motion and 3D Structure from Image Sequences

Our previous work on registration and estimation from long image sequences is described in [21, 22]. [7] addresses the problem of estimating the structure and mo-

tion of a smooth curved object from its silhouettes observed over time by a trinocular stereo rig under perspective projection. We first construct a model for the local structure along the silhouette for each frame in the temporal sequence. Successive local models are then integrated into a global surface description by estimating the motion between successive time instants. The algorithm tracks certain surface features (parabolic points) and image features (silhouette inflections and frontier points) which are used to bootstrap the motion estimation process. The entire silhouette along with the reconstructed local structure is then used to refine the initial motion estimate. We have implemented this approach and applied it to real images.

We have investigated the use of our earlier work on analysis-guided-synthesis for augmented reality environments. This involves integrated estimation of 3D motion and the orientation of a planar surface, and the use of the estimates to select and display a subset of image features that depict the estimated motion and structure. Additional, synthetic features can also be included to augment the set of selected original features [12].

We have reviewed the use of 2D and 3D motion for video compression and coding. The introduction of the MPEG-4 proposal has motivated a wide variety of approaches aimed at achieving a new level of video compression for very low-bit rate coding. We have divided the progress in very low-bit rate coding into three main areas: (1) waveform coding, (2) 2D content-based coding, and (3) model-based coding [52]. We have reviewed some common questions in image and video coding in [53], with focus on the estimation of 2D and 3D motion for use in efficient compression of video sequences using 3D models.

5.3 Nonrigid Motion

Our work on nonrigid motion has focused on the case of human motion analysis. Modeling spatial-temporal articulation patterns is very important for realistic rendering of face images in applications such as talking face and intelligent computer agents. We have developed a new approach to analyze, encode and learn human facial movement patterns [43, 44]. The approach consists of three major parts: spatial demension reduction through principal component analysis; so called thread method which approximates the temporal variation using simple basis functions; and learning to improve recognition and compresison capability. This scheme is also used for the compression of parameter sequences corresponding to facial articulation. Though developed based on MPEG4 facial animation parameter set, the algorithm can be easily applied to other parameter representations. A bit rate of 0.5Kb/sec is obtained for sequences of medium facial activities [45]. A facial expression recognition algorithm using recurrent neural network is investigated. The inputs to the network are the most significant components of this new data representation. Experimental results show that computational complexity is reduced and expressions can be correctly recognized even with different sampling rates.

[48, 49] present an algorithm for automatic head tracking using a model-based approach. The input is a 2D video sequence of a head-and-shoulders scene and the output is the trajectories of salient facial features, as well as an estimate of the 3D motion of the head. We consider feature tracking in two main steps: (1) Estimation of rigid head motion and (2) Non-rigid facial feature tracking. Localization inaccuracy and error accumulation are overcome by using an underlying 3D model to compute optimal templates for each video frame for use in the feature tracking module. Feature tracking is performed using a Bayesian-net assisted SSD framework and compensation of non-frontal views using the estimated 3D motion. For local tracking algorithm, a probabilistic framework is used and related 2D distribution parameters are derived through training data. The network contains high level structural information about the face feature locations. A 3D head model, head pose estimation, and texture mapping are used to produce accurate templates for matching in the feature tracking module [50, 51]. In this way, the template database is constantly updated and can accommodate a large range of head motions without loss of precision. The initial feature identification is performed automatically and the tracking is successful over a large number of video frames. Computational complexity is also considered with the aim towards creating a real-time end-to-end model based video coding system.

6 Representation and Recognition

We have made progress in the representation of both 2D and 3D data. In the 2D case, we have derived the medial axis transform of image regions [20]. Instead of using the shortest distance to the region border, a potential field model is used for computational efficiency. The region border is assumed to be charged and the valleys of the resulting potential field are used to estimate the axes for the medial axis transform. The potential valleys are found by following force field, thus, avoiding two-dimensional search. The potential field is computed in closed form using the equations of the border segments. The simple Newtonian potential is shown to be inadequate for this purpose. A higher order potential is defined which decays faster with distance than as inverse of distance. It is shown that as the poten-

tial order becomes arbitrarily large, the axes approach those computed using the shortest distance to the border. Algorithms are given for the computation of axes, which can run in linear parallel time for part of the axes having initial guesses.

For the 3D case, [42] presents an algorithm for reducing a set of high-density scanned range data to a simplified polygonal mesh. Of major interest is the application of this algorithm to Cyberware 3D range data of human heads which produces simple yet accurate wireframe approximations. The objective is to decimate the range data while maintaining acceptable levels of resolution over critical sections of the face such as areas of high curvature (nose, mouth) and sections with fine detail (eyes). Areas such as forehead and cheeks which are relatively smooth are represented with lower detail. In an application of 3D models, we have used the octree representation for collision detection among moving 3D objects [17].

For recognition, we have developed an approach to learn the low-level image structure of a class of objects from observations of many samples of the class [38]. Canonical, multiscale intensity patterns are learned from sample gray-level images. The gray-scale regions are obtained from the multi-scale segmentation algorithm described earlier [19]. However, there are inherent difficulties in obtaining an optimal set of segmented regions for pattern recognition purposes, which motivates blending of segmentation and image interpretation. The canonical representation is extracted at different scales using a neural network learning algorithm. Regions at a range of scales are extracted and examined for merger to obtain largest homogeneous components. These merged regions at each scale are then matched across the tree descriptions of sample images. Learning is based on regions properties, such as shape, area, gray-level intensity as well as their spatial relationships. This enables the network to extract region based descriptions at several scales which constitute a cononical representation of the object and can be used to recognize the object. In our experiments, we have used images from a face database. This results in a facial description in terms of prominent gray-scale facial features such as eyes and mouth. Our earlier work on learning of 2D object models appears in [23].

References

[1] N. Ahuja and T. Huang, IU at UI: An Overview and an Example on Shape from Texture, *Proc. DARPA Image Understanding Workshop*, Boston, pp. 222-253, April 6-8, 1988.

[2] N. Ahuja, IU at UI: An Overview of Research during 1988-90, *Proc. DARPA Image Understanding Workshop*, Pittsburgh, pp. 134-140, Sep 11-13, 1990.

[3] N. Ahuja and T. S. Huang, IU at UI: An Overview of Research During 1990-91, *Proc. DARPA Image Understanding Workshop*, San Diego, pp. 127-135, Jan. 27-29, 1992.

[4] N. Ahuja and T. S. Huang, IU at UI: An Overview of Research During 1991-92, *Proc. DARPA Image Understanding Workshop*, pp. 117-125, April 1993.

[5] N. Ahuja and T. S. Huang, IU at UI: An Overview of Research During 1993-94, *Proc. ARPA Image Understanding Workshop*, pp. 133-142, November 1994.

[6] N. Ahuja and T. S. Huang, IU at UI: An Overview of Research During 1994-95, *Proc. ARPA Image Understanding Workshop*, pp. 159-164, February 1996.

[7] T. Joshi, N. Ahuja and J. Ponce, Structure and Motion Estimation from Dynamic Silhouettes, submitted.

[8] N. Ahuja and J-H. Chuang, Shape Representation Using a Generalized Potential Field Model, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 19, No. 2, pp. 169-176, February 1997.

[9] K. Bowyer and N. Ahuja, *Advances in Image Understanding: A Festschrift for Azriel Rosenfeld*, IEEE Computer Society Press, June 1996.

[10] N. Ahuja and R. Charan, Pixel Matching and Motion Segmentation in Image Sequences, *Lecture Notes in Computer Science - Recent Developments in Computer Vision 1035*, S.Z. Li, D.P. Mital, E.K. Teoh, and H. Wang (eds.), Springer-Verlag, pp. 139-148, 1996.

[11] N. Ahuja, A Transform for Multiscale Image Segmentation, K. Bowyer and N. Ahuja (eds.), *Advances in Image Understanding: A Festschrift for Azriel Rosenfeld*, IEEE Computer Society Press, pp. 45-64, June 1996.

[12] N. Ahuja and S. Sull, Analysis Guided Video Synthesis for Hyper Reality, N. Terashima and J. Tiffin (eds.), *Hyper Reality: The Infrastructure of the Information Society*, to appear.

[13] A. Krishnan and N. Ahuja, Range Estimation from Focus using a Nonfrontal Imaging Camera, *Int. Journal of Computer Vision*, Vol. 20, No. 3, pp. 169-185, 1996.

[14] N. Ahuja, On Detection and Representation of Multiscale Low-Level Image Structure, *ACM Computing Surveys*, Vol. 27, No. 3, pp. 304-306, September 1995.

[15] T. Srinkanth and N. Ahuja, Parallel Distributed Detection of Feature Trajectories in Multiple Discontinuous Motion Image Sequences, *IEEE Trans. on Neural Networks*, Vol. 7, No. 3, pp. 594-603, May 1996.

[16] S. Das and N. Ahuja, Active Surface Estimation: Integrating Coarse-to-fine Image Acquisition and Estimation from Multiple Cues, *Artificial Intelligence*, Vol. 83, pp. 241-266, 1996.

[17] Y.Kitamura, H. Takemura, N. Ahuja and F. Kishino, Colliding Face Detection Among 3-D Objects Using Octree and Polyhedral Shape Representation, *Journal of the Robotics Society of Japan*, Vol. 14, No. 5, 7, pp. 733-742, 1996.

[18] M. Tabb and N. Ahuja, Unsupervised Multiscale Image Segmentation by Integrated Edge and Region Detection, *IEEE Transactions on Image Processing*, May 1997, to appear.

[19] N. Ahuja, A Transform for Multiscale Image Segmentation by Integrated Edge and Region Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 18, No. 12, pp. 1211-1235, Dec. 1996.

[20] N. Ahuja and J-H. Chuang, Shape Representation using a Generalized Potential Field Model, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 19, No. 2, pp. 169-176, Feb. 1997.

[21] C. Debrunner and N. Ahuja, Segmentation and Factorization-Based Motion and Structure Estimation for Long Image Sequences, *IEEE Trans. Pattern Analysis and Machine Intelligence*, to appear.

[22] J. Weng, Y. Cui and N. Ahuja, Transitory Image Sequences, Asymptotic Properties, and Estimation of Motion and Structure, *IEEE Trans. Pattern Analysis and Machine Intelligence*, to appear.

[23] J. Weng, N. Ahuja and T. S. Huang, Learning Recognition and Segmentation Using the Cresceptron, *Int. Journal of Computer Vision*, to appear.

[24] D. Hougen and N. Ahuja, Shape from Appearance: A Statistical Approach to Surface Shape Estimation, *Proc. European Conf. on Computer Vision*, Cambridge, England, pp. 421-429, April 1996.

[25] A. Krishnan and N. Ahuja, Panoramic Image Acquisition, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, pp. 379-384, June 18-20, 1996.

[26] P. Bajcsy and N. Ahuja, Uniformity and Homogeneity-based Hierarchical Clustering, *Proc. Int. Conf. on Pattern Recognition*, Vienna, Austria, pp. 96-100, Aug. 26-29, 1996.

[27] S. Jackson and N. Ahuja, Elliptical Gaussian Filters, *Proc. Int. Conference on Pattern Recognition*, Vienna, Austria, pp. 775-779, Aug. 26-29, 1996

[28] T. Courtney and N. Ahuja, Segmentation of Volume Images using a Multiscale Transform, *Proc. Int. Conf. on Pattern Recognition*, Vienna, Austria, pp. 432-436, Aug. 26-29, 1996.

[29] K. Ratakonda and N. Ahuja, Segmentation Based Reversible Image Compression, *Int. Conf. on Image Processing*, Geneva, Switzerland, pp. 84-84, Sept. 1996.

[30] K. Ratakonda and N. Ahuja, Discrete Multi-Dimensional Linear Transforms over Connected Arbitrarily Shaped Supports, *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1997, to appear.

[31] P. Bajcsy and N. Ahuja, A New Framework for Hierarchical Segmentation using Homogeneity Analysis, *Proc. First Int. Conf. on Scale-Space in Computer Vision*, 1997, to appear.

[32] N. Srinivasa and N. Ahuja, Learning to Fixate on 3D Targets with Uncalibrated Active Cameras, 1997, *these proceedings.*

[33] P. Bajcsy and N. Ahuja, Segmentation of Multivariate Multidimensional Functions, *IEEE Trans. on Image Processing*, No. IP-1445, 1997, submitted.

[34] J. Ma and N. Ahuja, Recovering Dense Structure from Motion by Region Correspondence, *Proc. ICIP-97*, 1997, submitted.

[35] P. Bajcsy and N. Ahuja, Heirarchical Anisotropic Segmentation Using Attribute Dictionary, to be submitted.

[36] M. Tabb and N. Ahuja, A Multiscale Region-Based Approach to Pixel Flow Estimation, to be submitted.

[37] K. Ratakonda and N. Ahuja, Resolution Enghancement with Adaptive Interpolation, *ICIP-97*, submitted.

[38] B. Perrin, Learning Feature-based Geometric Models of Objects from Images, M.S. Thesis, University of Illinois, 1997.

[39] P. Bajcsy and N. Ahuja, Hierarchical Image Segmentation Using Similarity Analysis, *these proceedings.*

[40] K. Ratakonda and N. Ahuja, Super Resolution With Region Sensitive Interpolation, *these proceedings.*

[41] R. J. Qian and T. S. Huang, Optimal Edge Detection in Two-Dimensional Images, *IEEE Transactions on Image Processing*, Vol. 5, No. 7, pp. 1215-1220, 1996.

[42] Ricardo Lopez and Thomas S. Huang, Simplification of 3D Scanned Head Data for Use in Real-time Model Based Coding Systems, *SPIE Visual Communications and Image Processing*, Orlando, pp. 218-227, FL, March 1996.

[43] H. Tao and T. S. Huang, Motion patterns in face animation, *IJCAI Workshop on Animated Interface Agents: Make Them Intelligent*, Nagoya Japan, August 1997, submitted.

[44] H. Tao and T. S. Huang, Modeling spatial-temporal patterns in facial articulation, *IEEE Workshop on Nonrigid and Articulated Motion*, June 1997, submitted.

[45] H. Tao, H. Chen and T. S. Huang, Analysis and compression of facial animation parameter set (FAPs), *IEEE Workshop on Multimedia Signal Processing*, June 1997, submitted.

[46] H. Tao and T. S. Huang, Color Image edge detection using cluster analysis, *IEEE Int. Conf. Img. Proc. (ICIP'97)*, October 1997, submitted.

[47] H. Tao and T. S. Huang, Multi-scale image warping using weighted Voronoi diagram, *IEEE Int. Conf. Image Processing*, Switzerland, 1996.

[48] R. Lopez, A. Colmenarez, T. S. Huang, Head and Feature Tracking for Model-based Video Coding, *International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging*, Rhodes, Greece, Sept. 1997.

[49] H. Tao, R. Lopez, and T. Huang, Facial Feature Tracking Under Varying Pose Using Bayesian Nets, *International Workshop on Coding Techniques for Very Low Bit-rate Video*, Linkoping, Sweden, July 1997.

[50] A. Colmenarez, R. Lopez, and T. Huang. 3D Model-Based Head Tracking, *Visual Communications and Image Processing*, San Jose, CA, SPIE Press, Feb. 1997.

[51] T. Huang, R, Lopez and A. Colmenarez, Feature Tracking using Model-based Pose Estimation, *5th International Workshop on Time-Varying Image Processing and Moving Object Recognition*, Florence, Italy, September, 1996.

[52] T. Huang J. Stroming, Y. Kang, and R. Lopez, Advances in very low bit rate coding in North America, *IEICE Journal*, Special Issue on Very Low Bit Rate Video Coding, Japan, IEICE Press, October 1996.

[53] T. Huang and R. Lopez, Computer Vision in Next Generation Image and Video Coding, Recent Developments in Computer Vision, *Lecture Notes in Computer Science*, Springer-Verlag, pp. 13-22, January 1996.

Image Understanding Research at Hughes Aircraft Company: Adaptive Image Exploitation

David M. Doria
Hughes Aircraft Company
P.O. Box 902, El Segundo, Ca 90245
EO/E1/A172
doria@warp10.es.hac.com
http://www.cerf.net/hac/products/atdr.htm

Abstract

In this paper we present an overview of plans for research under the image exploitation (IMEX) area of DARPA's image understanding program[1]. The primary goal of the planned research is the development of image restoration and feature extraction techniques, in combination with a performance model, for an adaptive model based matching system. This will enable improved and ideally optimal adjustment of the feature operators in terms of detectability of signal versus noise and feature spatial resolution, with optimality expressed in terms of overall system level probability of detection and probability of identification versus false alarm rate. The automatic target detection system is based on an adaptation of the Hausdorff metric to model based matching [Huttenlocher et. al. 1993]. The investigation will cover a variety of edge and feature operators, and analyze performance of the system using both linear restoration and superresolution methods, with the goal of achieving large improvements in the measured performance of the system.

1 Introduction

This program will develop methods of adaptive control of the automatic target detection and recognition (ATD/R) systems by means of a mathematical performance model that predicts the performance of the system in terms of the probability of detection (Pd), probability of identification (Pid), and false alarm rate (FAR). Both component level and end-to-end performance will be modeled.

We believe that it is unrealistic to expect image analysts or users of ATD/R's to be extremely expert in the tuning of such systems; nor is it desireable that re-tuning be needed on a scenario by scenario basis. Thereby is the motivation for an adaptive system that adjusts its parameters and/or thresholds to achieve useable performance specifications.

The goal of the adaptive system is to be able to relate the internal free parameters of an automatic target detection and recognition system (ATD/R) to the expected performance of the system in terms of the Pd, Pid, and FAR. One of our key research goals is to relate early feature extraction operators and thresholds, in addition to later stage model based search parameters, matching thresholds and geometric tolerances, to the expected system performance. The extraction of image features used by the model based system will be optimized by combining image restoration and feature extraction processes. We will also apply the same methods to the identification problem to optimize system parameters in terms of discriminability of a limited set of hypothesized target types and states.

[1]This work is supported by DARPA under Air Force program F33615-97-C-1022.

1.1 Performance Modeling

This effort will leverage off progress in the area of performance modeling for automatic target recognizer systems under a previous image understanding effort by Hughes Aircraft Company and Cornell University [Doria and Huttenlocher 1996, Doria 1996, Doria 1997], and builds on work by Grimson and Huttenlocher [1994]. Our goal in the analysis to date has been to arrive at a first order end-to-end model of the Pd and FAR, and to be able to relate these to predict expected ROC's as a function of local image complexity (among other parameters). The ATD performance model developed by Hughes and Cornell models the expected probability of detection Pd and the expected false alarm rate FAR for a given local part of a scene. The detection model predicts the Pd as a function of the atmoshperic attenuation, target versus background temperature difference, range to target, target size, sensor noise, sampling, and blur, edge operator, and matching algorithm tolerances and thresholds. The system models the match quality of a fixed-distance and measured fraction Hausdorff matching approach, as a function of the observed target features (in this case edges). The match statistic is the fraction of the model matched to the data, and the probability of this fraction being observed is the outcome of the model. The FAR model also includes terms that describe the the effects of search area, background clutter density and correlation, and correlation between target models. An estimate of the local probability of a false alarm P_{FA}in a region surrounding each pixel is obtainable. An example of an initial implementation of the P_{FA} estimate at regions surrounding each pixel is shown in Figure 1. Because the Pd and FAR are both functions of the scene, sensor, and algorithm parameters [2], these can be related to each other and a predicted ROC curve generated.

2 Objectives

The overall objective of this research is the development of a model driven ATD/R module that is able to adapt itself to different scenarios such that the overall performance of the system is optimized in terms of some specified criterion. One of the key benefits of this type of model is that it allows trade-offs with parameters describing each of the modeled elements of the system, and contributes to a general understanding of the relationship between these parameters in terms of performance. The performance model accepts a set of high level specifications such as Pd and FAR from the user, and translates these, by means of a criterion function, into ATD/R algorithm parameters. The initial performance model describes the trade-off between the probability of detection and false alarm rate for a class of algorithms that use the Hausdorff metric for matching geometric edges to model contour and/or internal detail thermal edges [Grimson and Huttenlocher 1994, Doria 1996]. An initial version of an adaptive ATD has been coded and tested on real FLIR imagery [Doria and Huttenlocher 1996, Doria 97].

One of the central goals of the present program is modeling the geometric and statistical behavior of low level feature operators, and developing an optimization strategy that incorporates the expected performance of the feature operators on portions of the scene. By applying a spatial restoration operation in combination with the feature operator, the expected feature signal response versus noise response can be predicted. The response of the feature operators on clutter and on target edges will be modeled. We plan to select and analyze some well known feature operators such as the Marr-Hildreth zero crossings of Laplacian of Gaussians [Marr and Hildreth 1980], directional derivatives of Gaussians, and the Canny operator [Canny 1986]. Both the spatial resolution and statistical performance of these operators will be modeled, and related to the performance of the ATD/R system.

Image restoration methods are well known, and in general are applied to measured imagery to recover the information in an image prior to the effects of sensor noise, blur, and other distortions such as motion, rotation, space-varying distortions, etc. It is a relatively common observation that the performance of automatic image exploitation and ATD/R systems is a strong function of image resolution and noise, which in turn are functions of the range to target, atmospheric blur, target contrast, and sensor blur and sampling effects. Torre and Poggio noted that detection and localization of a step edge are optimized by edge operators of different ex-

[2]Note, however, that some parameters are unique to both models; e.g. the FAR is not a function of actual target contrast

tent [Torre and Poggio 1986], which corresponds to a trade-off between resolution and noise sensitivity. By combining a performance model of the behavior of a given ATD/R algorithm(s) with a geometric/statistical model of the feature operators, the amount of resolution recovery versus allowed noise amplification that optimizes overall system performance can, in principle, be achieved. This can be done as a function of local image complexity, so that the algorithm operates both at the feature extraction level and the matching level with a "best" set of parameters, which are arrived at by an on-line optimization.

For linear convolution-type feature operators, it is possible to combine an image restoration operation with the feature operator by multiplying their transfer functions in frequency space. We plan to initially study the use of a parametric Weiner filter (PWF) or parametric goemetric Weiner filter (PGWF) [Castleman 1979] for estimation of target edges versus system noise and background clutter. The optimization can be carried out by varying the free parameter(s) of the PWF/PGWF, the edge/feature thresholds and parameters, and ATD/R search and matching parameters. Note than because the optimization is over both geometric tolerance and target versus clutter and noise responses, and also includes non-linear effects resulting from the edge/feature thresholds and the subsequent search and model matching terms, the minimum Bayes error at the feature extraction level is not necessarily optimal for the ATD/R problem. The system level optimization will attempt to find the globally best set of parameters, within bounding constraints, with optimality defined for all parameters in terms of the user level Pd, Pid, and FAR specifications.

The existing performance model makes use of estimates of the probability of detection of both target and background edges. The background edge estimation will be studied in combination with various models of local clutter spatial statistics. Prior knowledge or local estimates of the image and target power spectra will be used. Based on this information, an optimization, initially using a conjugate gradient based approach, will be performed. Note that this optimization occurs prior to any actual edge or feature extraction or model matching taking place. Table 1 gives a list of the parameters that will be involved in the search/optimization process. A combined restoration and feature extraction operation is applied only once to the measured image, based on the results of the optimization. Thus the adaptive system need not iterate over actual ATD/R results, but only over expected results as predicted by the model. As a result, the ultimate realizable throughput of the adaptive performance-model-driven system is not expected to be significantly limited by the optimization stage.

Table 1: Table of parameters subject to optimization.

Edge operator spatial frequency response.
Edge operator threshold.
Edge operator spatial tolerance.
Other feature parameters.
Model pose search volume.
Feature model-to-data match error tolerance.
Match quality threshold at a fixed distance, or match error at a quantile of feature matches.
Possible target fractional occlusion or non-observability.
Parametric Weiner filter parameters.

As an element of the modeling effort, we will investigate statistical models of both the target and background. Targets will be modeled in terms of their overall thermal contrast and the statistical variability of contrast. Background complexity models will be used to capture the local structure and statistical properties of the scene, and to relate these properties to the expected performance of the algorithms. This has already been done to a first order in the existing performance model, where targets have been modeled as rectangles of constant ΔT relative to the background. A fractional partial observability term has also been used to model those cases where targets are either partially occluded or have very low contrast over a portion of their contour. A first order Markov model has been used to model the correlation between background edges.

2.1 Optimization

One primary mode of optimization will be to operate in a Neyman-Pearson mode, where a FAR specification is given and the system optimizes Pd with respect to the free parameters in the restoration filter, feature extraction operators, and the performance model of the ATD/R. Over a limited range of system parameters, we envision that FAR iso-

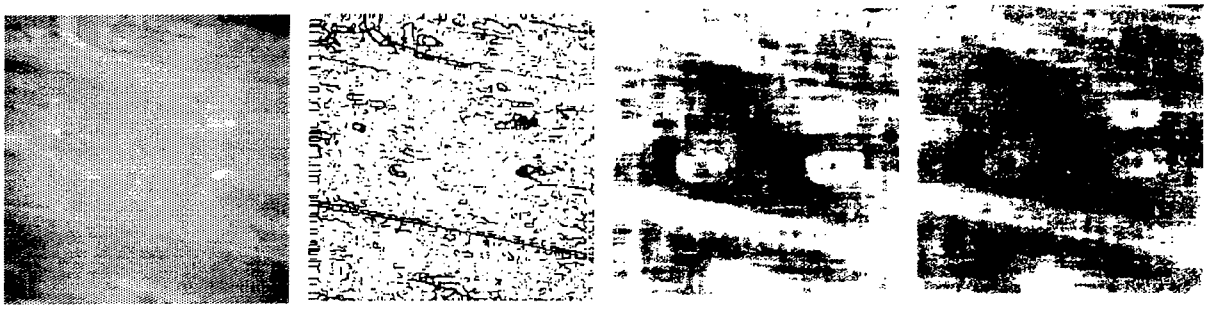


Figure 1: Example of P_{FA} images. Left to right: Original FLIR image, edge image, P_{FA} at match threshold of .80, P_{FA} at match threshold of .95. White in the P_{FA} images indicates high probability of false alarm in the local area.

contour lines of the set of parameters will be be generated, where each point on the iso-contour line is associated with a set of parameters that give equivalent expected FAR. The Pd that is maximized over these iso-contour lines in parameter space is found, and the parameters associated with the maximal Pd are then used by the system in actual processing.

2.2 Evaluation

We will perform evaluations of selected real FLIR data, measure the performance of the ATD/R system, and estimate the relative benefits of the adaptive versus non-adaptive systems. We will study the relative benefits of optimizing at the several levels of the ATD/R algorithms. Full ROC curves will be obtained experimentally. Estimates of the confidence intervals of the results will also be reported. When scene conditions change, for this (and all) ATD/R's there is a consequent change in the actual ROC curve of the system. Thus, we are interested in estimating system performance as a function of scene conditions. For a known sensor and database, results of the algorithms will be obtained as a function of range-to-target, image complexity, target set, target contrast, and sensor resolution and noise parameters. Results of the Pd corresponding to low, medium, and high values of the specified FAR will also be obtained.

2.3 Use of Prior Knowledge

Initially, we will make use of a limited amount of prior knowledge, and determine the performance of the system. Among the initial categories of knowledge will be the sensor modulation transfer function (MTF), noise characteristics and $NE\Delta T$, and sampling properties. We will also apply local clutter models if these are available from a focus of attention module. The system will estimate the background clutter density and correlation properties. Clutter models will be studied in terms of their utility for performance characterization with the algorithms being used, and their tractability for analysis. Expected target types and frequencies of targets will be used to constrain the target hypotheses, and to act as Bayesian priors on the system processing.

We will then proceed to add other information, such as that available with the use of site models. Where a site model allows estimation of the location of a sharp intensity transition, the total system blur will be estimated from the observed edge spread function. In addition, background models may provide improved estimates of local scene clutter characteristics. Prior estimates of target types and poses may also allow estimates of the target power spectra, thereby improving an image restoration process.

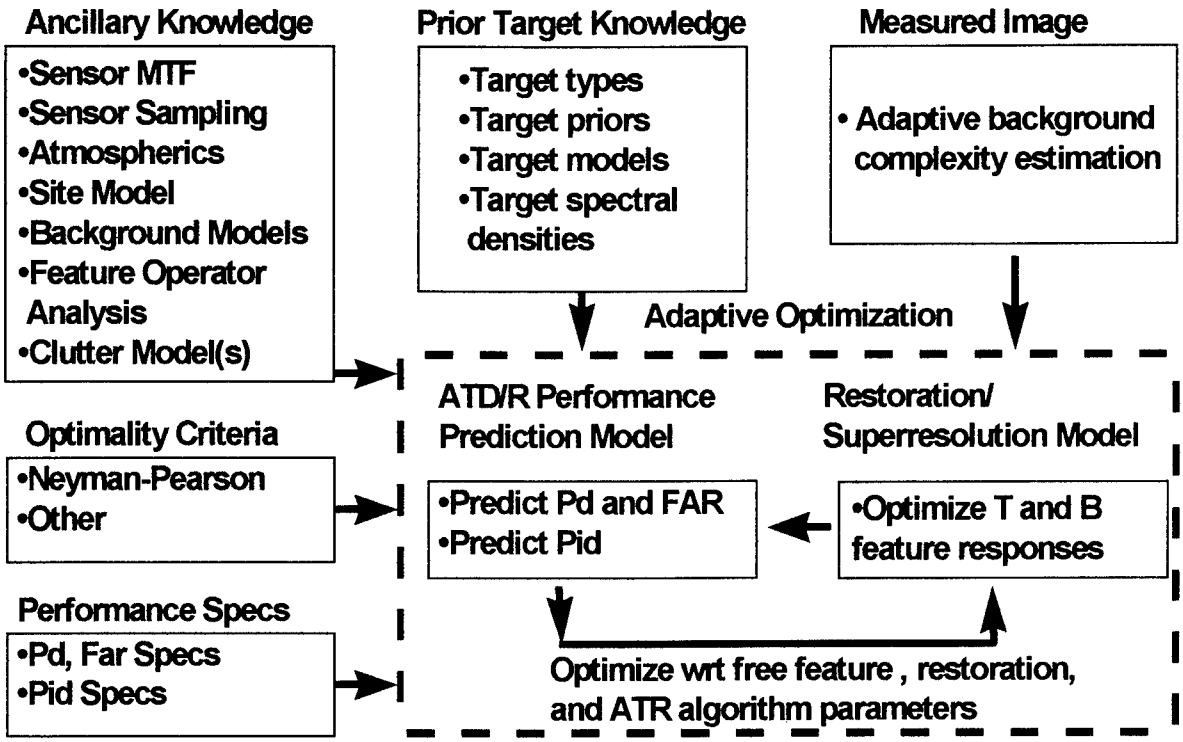


Figure 2: Overview of high level optimization and use of prior knowledge (T=target, B=background).

2.4 Extension to Probability of Identification Optimization

Thus far, the performance analysis of the Hausdorff-type, geometric model-based algorithms has not been extended to the modeling of the performance of the system at the recognition or identification levels; that is, discrimination between a limited number of competing target hypotheses. If a hypothesis includes target type, configuration, and pose, the problem becomes one of discriminating between objects based on their quality of match to the data. As part of the performance modeling effort in this program, we will extend the existing performance analysis to handle the target discrimination problem. As a general trend, we expect that the spatial resolution versus noise trade will result in a different "best" set of feature parameters for the identfication problem than those required for the corresponding detection problem. The model set is assumed constant for the detection stage, but is a function of the earlier indexing and search/match stages at the identification stage. In this analysis, a Neyman-Pearson criteria no longer is applicable, therefore we will initially apply a Bayes risk criteria. Given a defined set of target hypotheses, the goal of the extended performance model is to predict the performance of the model based system as a function of the operating parameters of the entire sensor-ATR chain of measurement and processing. As with the detection-level performance model, the system will predict performance as a function of the edge/feature operator parameters and thresholds, and the quality-of-match criteria of the models to the data.

2.5 Extended Features

We propose to augment the initial edge based matching methods based with additional local features. To this point we have discussed the ATD/R system and performance model in terms of performance on geometric information contained in image and model edges. The edge extraction process is included in the performance model, and hence actual grey level statistics are in fact related to expected Pd and FAR. It has been shown that the use of orientation information has the potential to improve the discriminability of target from clutter [Olson et.

al. 1996]. We expect that this will also hold when applied to the target hypothesis discrimination case.

Wavelets are particularly attractive for this investigation due to their natural expression in terms of varying scales and associated spatial frequencies. The results of the scale and frequency versus noise analysis of selected wavelet types will be used in a completely analogous manner within the ATD/R as the analysis of the edge operators. The ATD/R match quality metric may need to be modified to include additional parameters; however, this will be true even for the addition of oriented edges to the match quality statistic. This effort will, we believe, extend the utility of the present analysis based adaptive system to a more general and expanded set of model based matching approaches and algorithms, and also suggest which, of a set of low level features, are best for the respective detection and identification problems.

2.6 Superresolution

Given the introduction of additional knowledge about the scene or object that is being observed, it may be possible to recover spatial frequencies above the diffraction limit [Hunt 1995]. Potential types of knowledge about targets include spatial extent, non-negativity, target smoothness, and prior estimates of target power spectra. We will investigate promising approaches for superresolution such as the Poisson maximimum a-posteriori method of Sementilli et. al. [Sementilli et. al. 1993a], who have also derived an upper bound on superresolved resolution enhancement based on error tolerances within the recovered spectral range, the size of the observed object, the variance of the image noise, and the estimated error sidelobes [Sementilli et. al. 1993b]. Key issues that need to be resolved are the extent of resolution that is recoverable within a specified error, the noise model for the extended frequencies, which additional image and target constraints are most useful, and the convergence properties of the system. If the superresolution results on actual FLIR data appear to be useful based on an initial investigation, we will integrate the superresolution algorithm with the performance model and adaptive ATD/R system.

3 Summary

The key scientific and technical issues to be addressed in this effort are (a) the development and validation of useful mathematical models of ATD/R performance, (b) validation of the models both at the component and end-to-end levels, (c) estimation of the accuracy and extensibility of the performance models over new scenes, (d) development of combined restoration and feature extraction for optimal ATD/R performance, (e) development of optimality criteria at the feature level that are related to overall end-to-end system performance as defined by user specifications, (f) the Pd versus FAR performance improvement that is obtainable with the proposed adaptive ATD system, (g) the Pid performance obtainable with the extension of the performance model and adaptive ATR to the modeling of competing target hypotheses, and (h) the results of the utility of superresolution methods to FLIR ATD/R.

This modeling of the sub-components of the ATD/R system includes the sensor measurement process, feature extraction, search, and matching. Target and background models are also necessary and will be applied. The key trade-off in the modeling effort is that of mathematical tractibility versus performance prediction accuracy. A tightly integrated modeling and evaluation process is therefore necessary.

We envision the use of this system as a module level component within a larger image exploitation, force monitoring, or ATD/R system. Our goal is to very significantly improve the capability and robustness to new conditions of these systems, especially at extended ranges or less than perfect imaging conditions. The use of the adaptive model based development philosophy has, we believe, significant long term advantages over other general ATD/R and image exploitation approaches, as described in Table 2. While requiring additional validation of the performance model versus non-adaptive or non-analytic adaptive methods, the final result is a system that is capable of good performance over a wide range of operating conditions.

Further extension of the adaptive type of approach to include additional types of signatures is certainly feasible. The application in this program is primarily to FLIR and EO imagery, but the general mod-

Table 2: Characteristics of three classes of automatic target recognizer systems in terms of training (readiness) requirements, performance, and performance related to parameters. (SPR = statistical pattern recognition, MB = model based ATD/R, ADMB = adaptive model based ATD/R.)

ATR Class	Readiness Requirements	Performance	Performance related to parameters and scene content
SPR	Training with real data.	Excellent performance when test and training data are similar. Otherwise unpredictable (often poor).	No explicit relationships.
MB	Tuning with real data and models.	Good when test and training data are similar. Otherwise unpredictable.	May be explicit relationship. Point design of parameters.
ADMB	Tuning with real data and models. Validation of performance models required.	Good, robust to varying scene conditions. Performance model enables adaptivity.	Explicit relationships.

eling approach and adaptive system design can, in principle, be applied to other modalities, given the development of the proper target, sensor, feature, and algorithm models. The notion of a theory of ATR includes, we believe, this type of modeling effort, where progress can be defined in terms of insight gained into the problem, and predictions serve in a practical way to improve system capabilities.

References

[Huttenlocher et. al. 1993] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge,"Comparing Images Using the Hausdorff Distance, " IEEE PAMI, 15(9), 850-863, 1993.

[Doria 1996] D. M. Doria, "Geometric Model Based FLIR Performance Prediction," Proc. 5th. Science and Technology Conference on Automatic Target Recognition, Johns Hopkins APL, July 25, 1996.

[Doria and Huttenlocher 1996] D. M. Doria and D. P. Huttenlocher, "Progress on the Fast Adaptive Target Detection Program,"Proc. 1996 DARPA Image Understanding Workshop, Palm Springs, Ca., 1996, 589.

[Grimson and Huttenlocher 1994] W.E.L. Grimson and D. P. Huttenlocher, "Analyzing the Probability of a False Alarm for the Hausdorff Distance Under Translation," Proc. 1994 DARPA Image Understanding Workshop, 1994, 1257.

[Doria 1997] D. M. Doria, "Performance Modeling and Adaptive Target Detection," This proceedings.

[Castleman 1979] K. R. Castleman, "Digital Image Processing," Prentice Hall, 1979.

[Torre and Poggio 1986] V. Torre and T. A. Poggio, "On Edge Detection," IEEE Trans. Patt. An. and Mach. Intell. 8(2), 147, 1986.

[Marr and Hildreth 1980] D. C. Marr and E. C. Hildreth, "Theory of Edge Detection," Proc. Roy. Soc. London B, V. 207, 187, 1980.

[Canny 1986] . Canny, "A Computational Approach to Edge Detection," IEEE PAMI, 8(6), 679, 1986.

[Olson et. al. 1996] C. F. Olson, D. P. Huttenlocher and D. M. Doria, "Recognition by Matching with Edge Location and Orientation," Proc. 1996 DARPA Image Understanding Workshop, Palm Springs, Ca., 1996.

[Hunt 1995] B. R. Hunt, "Super-Resolution of Images: Algorithms, Principles, Perfor-

mance," Int. Journal or Imaging Syst. and Tech. 6, 297, 1995.

[Sementilli et. al. 1993a] P. J. Sementilli, M. S. Nadar and B. R. Hunt, "Poisson MAP super-resolution estimator with smoothness constraint," Proc. SPIE V. 2032, 2, 1993.

[Sementilli et. al. 1993b] P. J. Sementilli, B. R. Hunt and M. S. Nadar, "Analysis of the limit of superresolution in incoherent imaging," J. Opt. Soc. Am. A, 10(11), 2265, 1994.

Image Understanding Research at UC Riverside: Integrated Recognition, Learning and Image Databases

Bir Bhanu
College of Engineering
University of California, Riverside, California 92521
Email: bhanu@engr.ucr.edu
URL: http://constitution.ucr.edu

Abstract

This report summarizes the image understanding (IU) research being conducted at the University of California at Riverside (UCR) under the DARPA sponsored programs in learning, target recognition and image databases. The goal of our research is to develop robust, reliable and efficient algorithms and systems that can work effectively in real-world applications. The principal areas of investigation include physically-based approaches utilizing multiple representations for target detection and recognition using multisensor data, multistrategy learning-based approaches for IU, and image databases. Automatic target recognition, image exploitation, surveillance, dynamic multisensor databases are the principal applications areas of our research.

1 Introduction

The University of California at Riverside is conducting research in several different aspects of image understanding. We summarize the technical objectives and scientific issues of the new University Research Initiative effort just starting, and the important progress made in the areas of learning for image understanding and automatic target recognition using multisensor (SAR and FLIR) imagery during the period from November 1995 to March 1997.

2 Learning Integrated Visual Database for Image Exploitation

The DOD has critical needs for robust high performance automated systems that can recognize objects in reconnaissance imagery acquired under dynamically changing conditions and for systems that can efficiently extract information from enormous image databases. Our new research addresses two interrelated problems with the effectiveness and efficiency of automated/semi-automated techniques for image understanding. *First*, the lack of robustness in algorithms and systems for object recognition with changing environments. *Second*, the lack of scalable intelligent strategies for quickly extracting meaningful information from enormous, dynamically changing image databases. This project is distinguished from other image databases in the following areas: (a) The content-based image retrieval image database technology is used for designing reliable IU algorithms. (b) The system has learning capability, improving its performance with use, both in terms of processing speed and matching with the user's perception; (c) users can query the images as well as the processing algorithms; (d) an extensive amount of image-related information is stored for characterization of various features and algorithms.

We focus on the task of image exploitation. The operational goal is to monitor military forces (vehicles and equipment) in a small geographic area (10 Sq. miles) that *move, sit and then move.* This requires robust high performance IU systems for recognizing objects/events in multisensor imagery acquired under dynamically changing conditions, and efficiently extracting "information" from enormous dynamic databases and exploiting it to develop reliable IU systems that will adapt to changing environments. The results of this program will provide a significant tool for military and intelligence information systems that will directly contribute to meeting the DoD goal of dominant battlefield awareness.

Objectives: The overall scientific goal of this project is to demonstrate that the conjunction of learning, recognition, and content and context-based retrieval (CCBR) are necessary and sufficient for reliable IU. We believe that for the development of robust and reliable IU systems we need a new generation of IU research that integrates target recognition, learning and CCBR technologies. Each alone or any combination of two is not sufficient to develop reliable IU systems operating in dynamic real-world environments. We must combine them in an integrated system to develop the science for image recognition. The specific subgoals are:

*This work is supported by grants F49620-97-1-0184, F49620-95-1-0424 MDA972-93-1-0010 and DAAH049510049. The contents and information do not necessarily reflect the position or the policy of the U.S. Government.

(a) Techniques for adapting recognition algorithms and models to different theater of operations and target types.

(b) Algorithms for handling target configuration differences, articulations and occlusions without combinatorial explosion.

(c) Methods for database queries by example with multiple objects and relationships (semantic queries) for recognition of events or scenarios.

2.1 Adaptive Recognition Models for Different Environments

State-of-the-art image understanding (IU) algorithms and systems for image exploitation from SAR images generally use static algorithms. They possess no learning ability and cannot improve their performance with experience achieved over time. Since they possess no adaptive capability to adjust to varying sensor operational conditions (such as sensor differences, depression angles, and multiple polarizations) and deployment environments (such as desert, forest, agricultural, urban areas and their seasonal variations) they cannot migrate from one theater of operations to another. The objective of our research is an approach that applies adaptive learning algorithms to exploit context information and feedback on performance results to improve the performance of IU based force monitoring systems. We allow the image exploitation system to adapt itself to a variety of SAR clutter types and perform optimally under different operating conditions. The learning takes place to (a) adapt clutter models with changes in sensor operating conditions, (b) adapt classifiers for different clutter types, and (c) adapt parameters employed within feature groups based on target recognition results. The changes for different deployment environments such as forest, desert, jungle, arctic, etc.) are primarily reflected in the characteristics of the image background clutter. Thus, adapting IU system to varying clutters is of fundamental importance. The research contains the following innovative ideas:

Variety of Feature Groups to Build SAR Clutter Models: No single feature may capture all possible statistical/structural variations for different clutters involved in a SAR deployment environment. We use several groups of features based on (a) multiscale Gabor wavelet (b) self-similarity in natural scenes, (c) statistics of geometrical/structural elements, and (d) statistical features.

Learning Background Clutter Models Through a Supervised Self-Organizing Process: Instead of artificially assigning a distribution to clutter models, we build clutter models from examples through a supervised learning process. These clutter models are represented by compact self-organizing maps (SOMs) which capture the distribution of the training data without the need to store a large number of examples. The SOM technique is extended in our approach to an incremental supervised learning process for clutter characterization. We also use the self-organizing map to classify a given region of an image into a clutter or a target area. The classification algorithm is adapted for different clutter types.

Stochastic Reinforcement Learning Technique to Adapt Clutter Models to SAR Sensor Operating Conditions: Different Sensor operating conditions correspond to varying weights of different feature groups which together constitute a model for a particular clutter type. The relationship between operating conditions and the weight of feature groups is optimized through a stochastic reinforcement learning process. This learning paradigm is used here since the human supervisor (man-in-the-loop) will only be tell the system that it is doing a "good job" rather than helping the computer in finding the association between operating conditions and the weights of different features.

Delayed Reinforcement Learning for Learning Clutter Model Parameters Based on Target Recognition Results: The image exploitation process requires a sequence of algorithms for CFAR (Constant False Alarm Rate) detection, feature extraction, clutter characterization and target recognition. It is inherently a multi-stage process that has delay from stage-to-stage. Since we cannot determine the goodness of different stages until we have seen the final recognition result, it is natural to evaluate the quality of earlier stages based on the final recognition results and a delayed reinforcement learning technique fits this situation exactly.

2.2 Algorithms for Handling Articulations and Occlusions

Current methods for target recognition in SAR imagery cannot handle target articulation, configuration differences or moderate occlusion. The objective of the research is to focus on the challenging problems caused by target variations due to articulation or configuration differences. Our approach to the problem of automatic model construction and recognition of articulated, non-standard targets in SAR imagery is based on local features and local reference coordinate systems. We have a systematic method for constructing recognition models of objects that are not articulated and then we employ local image features to match these models and recognize the same objects in articulated positions or non-standard configurations. The key features of the approach are:

Sensor Specific Design Approach for SAR Target Recognition: The unique characteristics and physics of SAR sensors are recognized and accommodated

by our design approach. The natural range/cross-range coordinates and tessellation are directly incorporated. The translation invariance is captured by using relative positions of SAR specific features and the large rotational variances are accommodated by modeling an appropriate number of azimuths.

Models Based on Articulation Invariants: Our approach for SAR target recognition makes use of the existence of articulation invariants. The models are stored for standard non-articulated objects. Thus, it avoids the combinatorial explosion of model configurations and is inherently directly applicable to matching the un-occluded regions (of occluded objects).

Physically Based Local SAR Image Features Accommodate Articulation: The relative distances between scattering centers (and other features such as topographic primal sketch features, reflector geometry, feature sequences based on location and relative amplitude, and polarization based features) are related to the shape and physical dimensions of the detailed target geometry. The local coordinate approach to local features (vs. global approaches or even a local neighborhood approach) accommodates articulation/occlusion without precluding use of widely separated features (which are good discriminators).

Efficient Search for Positive Evidence is Designed to Accommodate Spurious Data: A powerful combination of a true look-up table and a voting technique that searches for positive evidence reduces the work on all non-matching cases to the random coincidences and makes the method scale gracefully.

Super-Resolution Target Chips: Super-resolution (e.g. six inch) provides rich feature sets that allow matching the non-articulated or un-occluded regions of the target. Since it is not clear that the problem is solvable at one foot resolution, we have taken the unique approach of demonstrating feasibility at six inch resolution, and then investigating the performance degradation at one foot resolution real data.

Hierarchical Approach to Indexing and Matching for Handling the Exceptional Cases: The basic approach for indexing and matching based on the relative locations of HH-polarization signal strength maxima will be extended to other features (such as other polarizations and using the complex components) to handle the exceptional cases and additional matching modules, based on other features, will be applied to discriminate among ambiguous results. In addition, we explore a promising stochastic hidden Markov modeling (HMM) based approach for indexing/matching.

2.3 Database Semantic Queries for Recognition of Objects and Events

There are basically two approaches for searching image databases to identify objects. The first approach uses the traditional object recognition that requires the understanding of images. The second approach uses features for content-based retrieval to select images based on the chosen measure of similarity. It does not require the full understanding of images. We want to combine these two approaches for image exploitation application and investigate ways of using contextual data and domain knowledge for image interpretation. The key features of the approach are:

Flexible Similarity Measures and Indexing Functions: Current techniques for feature-based retrieval use a fixed set of features, similarity measures and indexing strategies that are determined in advance. We develop learning algorithms for feature selection, similarity metrics and associated indexing structures. We allow generation of run-time features and handle data and index management on the fly. We investigate techniques that permit efficient search of high dimensional space. This will allow improved performance in terms of retrieval speed and a quality of results approaching human perception of similarity.

In practice the relevance of each feature in classifying a new object may be different. In addition, the relevance of a feature may depend on the user and the object being classified. Inclusion of features with low relevance leads to high dimensionality of the feature vector and can degrade performance. What is required is to find the local relevance of each feature and use that information to define a flexible similarity measure that closely resembles human perception. Our approach for content-based retrieval is to learn the most salient features and develop flexible similarity measures that best resemble human perception of similarity for image exploitation.

Another important problem in large visual databases is the indexing structure to reduce the search space, allowing quick browsing. Since multiple features are normally necessary to represent an image, a multidimensional indexing structure is required. The performance of existing techniques for query by example critically depends on the selection of the features, the similarity measures, the user and the application context. In our approach the database is indexed by the order of the most significant factor/eigenfeature, the second most significant factor/eigenfeature, and so on. The query search in our approach consists of two stages: the pre-query stage and on-line stage. The learning is at two-levels: first to determine the local relevance of each feature, the ranking and selection of the features, and the indexing structure for the current user, query and application, and sec-

ond to select from the knowledge base the ranking of features and the indexing structure using the contextual information related with the application.

Data Models and Queries: We define a complete set of data models that the image database system is designed to handle. This includes data models for contextual information and the design of data structures for indexing and retrieval. There exist fairly complete data models for image formats and intermediate data types, which can be used as a basis of our development. The key issue is to devise an integrated model of images, image-related information, and processing algorithms.

We develop database access methods based upon content and context with common-sense and temporal reasoning capability, develop suitable query constructs and the semantics of linguistic constraints that allow one to express image-oriented queries, and designing an image data model that is sufficiently powerful, flexible, and extensible. Query methods can learn the selection of features and similarity measures that match with the user's perception, and the associated indexing structure. The learning approach will lead to improvement of performance with the use, both in terms of retrieval speed and user's perception of similarity.

Query language will perform associative search on images, features and algorithms. It will be sufficiently expressive and be capable of handling imprecise and incomplete data. What image resolution to use for query processing is an important optimization problem. We will investigate whether low-resolution intermediate results can be used to reduce the processing cost of image queries. Incomplete information often results in imperfect database schema, which need to evolve though learning, monitoring, or user overwriting. The schema may change at the data representation level (e.g. the attributes of an object, the class an object belongs to, etc.). It may also change at the conceptual level (e.g., the change of the class hierarchy). Both types of evolution will be studied.

Bayesian-Based Factor Analysis: Principal component analysis is a commonly used technique in image processing and has been recently used in visual databases. However, there are several limitations of this technique and the factor analysis model [28] has several important advantages: *first*, the factor analysis model permits a noise term, *second*, the factor analysis model postulates a linear model for the basic data vectors, and, *finally*, the factor analysis model is much more general, and is driven by a need to find and retain a meaningful correlation structure for the data that can be explained by a few linear combinations of some latent factors.

The method we develop and apply in this context involves scoring the image according to the Bayesian factor analysis model, which is ideally suited for image databases. It provides us a compact representation, contextual information for image exploitation can be explicitly accounted for in the model, and it is suitable for indexing, image recognition and classification.

Image Characterization: A variety of features are used in content-based retrieval for visible images. Many of these features are not useful for SAR, FLIR and multispectral images which are important for image exploitation. We develop image content based on wavelet (e.g. Gabor wavelet based representation has energy patterns that are localized both in the spatial domain and in the frequency domain) and information complexity measures (such as minimum description length) to characterize multisensor images.

2.4 Prototype System

Our new research will be build upon the Visual Intelligence Datablade system being developed by Virage Inc. This system [23] is based on a basic model called *Visual Information Management System (VIMSYS)* developed by Virage Inc. This model has four layers of information abstraction: the raw image, the processed image, the user features of interest and the user events of interest. The top three layers form the content of the image. There are mechanisms for defining and installing new similarity measures, called primitives. In addition, Virage has tools for graphical user interface, query canvas (query-by-sketch), light table (for displaying query result), and command line interface.

As part of the project we will develop algorithms and tools for image exploitation in the context of large databases. In addition, we will develop a research testbed to integrate image and context databases with both human customers and the target detection and recognition system algorithm customers.

2.5 Evaluation Plan

Our evaluation plan provides a significant emphasis on algorithm evaluation and will allow the subsystem technology developed to be evaluated in the context of overall system effectiveness.

The overall system performance metrics are a probability of detection (Pd) and a false alarm rate (FAR). Demonstration results for the clutter modeling will be expressed in terms of Pd and FAR, later results for recognition will be in terms of Pcc (probability of correct classification) and Pci (probability of correct identification). In addition, the performance of the learning system will be reported as a learning rate expressed in terms of performance versus the number of exemplars experienced. We plan to

use available SAR, visible and multispectral imagery and the imagery that may become available during the program and simulated SAR scenes produced by XPATCH at various depression angles and for different environments (e.g. forest, agriculture, desert shrub and desert) to populate the database.

The critical experiments are (a) demonstration of the capabilities of various feature groups and self organizing clutter models to distinguish man-made objects from natural clutter in actual SAR images and to show test results for scenes in simulated imagery. (b) the use of reinforcement learning to adapt the natural clutter models to sensor operating conditions. (c) learning rate (performance vs. experience) for retraining a clutter model with data from a different depression angle. (d) demonstrate the performance of clutter models that are adapted to new deployment environments (for example agriculture, and desert shrub) and report the learning rate results. (e) demonstrate the system level performance of the recognition elements integrated with the clutter models. (f) demonstrate the system level performance with the clutter models adapted to the matching results and also report both the learning rate and the point where learning transitions from supervised to unsupervised.

3 Multistrategy Learning for Image Understanding

The multistrategy learning-based IU approach selectively applies machine learning techniques at multiple levels of the IU process to achieve robust recognition performance. At each level, appropriate evaluation criteria are employed to monitor the performance and self-improvement of the system [5, 18].

With the goal of achieving robustness, our research at UCR is directed towards learning parameters, feedback, contexts, features, concepts, and strategies of IU algorithms for model-based object recognition. The progress made during the last year includes the following: (a) development of approaches based on reinforcement learning for controlling feedback between segmentation and recognition components in an object recognition system, and using it to learn segmentation and feature extraction parameters. (b) development of an approach based on reinforcement learning for integrating context with clutter models to reduce false alarms and improve target detection performance in FLIR images (c) development of a methodology to improve performance of an IU algorithm by adapting the input data into the desired form for a given algorithm. (d) development of a case-based reasoning approach for learning recognition strategies for image exploitation by categorization of images.

Earlier we have demonstrated the scalability of the genetic learning-based approach for adaptive image segmentation [12, 17]. We also developed basic ideas applicable to integrating information from multisensors or integrating recognition and motion analysis, using multiobjective optimization [2, 9].

3.1 Learning Recognition Strategies

We have developed several techniques for learning recognition strategies. These techniques are based on reinforcement learning and case-based reasoning.

3.1.1 Reinforcement Learning for Adaptive Algorithms, Parameters and Feedback in an IU System

Problem: To automate acquisition of recognition strategies in dynamic environments to develop theoretically sound approaches to control feedback which are based on the results of recognition and to learn segmentation and feature extraction parameters for robust model-based recognition.

Approach: We have developed two approaches based on reinforcement learning for closed-loop object recognition in a multi-level vision system. These approaches use the team of learning automata algorithm [26] and the delayed reinforcement learning algorithm [27].

The closed-loop object recognition system evaluates the performance of segmentation and feature extraction by using the recognition algorithm as part of the evaluation function. Recognition confidence is used as a reinforcement signal to the image segmentation or feature extraction processes. By using the recognition algorithm as part of the evaluation function, the system is able to develop recognition strategies automatically, and to recognize objects accurately in newly acquired images. As compared to the genetic algorithm [9, 10] which simply searches a set of parameters that optimize a prespecified evaluation function, here we have a recognition algorithm as part of the evaluation function [26].

In order to speed up the above algorithms we have developed a general approach [3] to image segmentation and object recognition that can adapt the image segmentation algorithm parameters to the changing environmental conditions. The edge-border coincidence is used for both local and global segmentation evaluation. However, since this measure is not reliable (see Figures 1 and 2) for object recognition, it is used in conjunction with model matching in a closed-loop object recognition system. Segmentation parameters are learned using a reinforcement learning algorithm that is based on a team of learning automata and uses edge-border coincidence or the results of model matching as reinforcement signals. The edge-border coincidence is used initially to

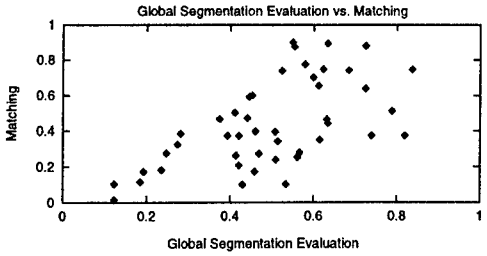


Figure 1: Global edge-border coincidence vs. matching confidence.

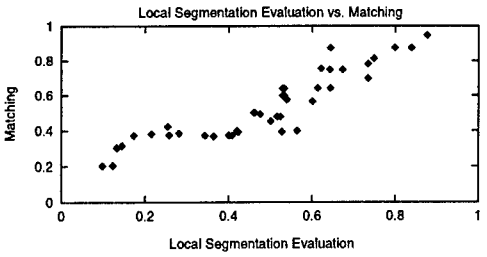


Figure 2: Local edge-border coincidence vs. matching confidence.

select image segmentation parameters using the reinforcement learning algorithm. Subsequently, feature extraction and model matching are carried out for each connected component which passes through the size filter based on the expected size of objects of interest in the image. The control switches between learning integrated global and local segmentation based on the quality of segmentation and model matching.

Accomplishments: Using the *Phoenix* algorithm for the segmentation of color images, a clustering-based algorithm for the recognition of occluded 2-D objects [11] and a *team of learning automata* [26] algorithm, or a *delayed reinforcement learning algorithm* [27], we show that in real images with varying environmental conditions and camera motion, effective low-level image analysis and feature extraction can be performed. We have shown performance improvement of an IU system combined with learning over an IU system with no learning [26, 27]. Figure 3 gives an example for performance improvement for both image segmentation and object recognition with experience. In this figure the traffic sign shown in the first column of images (taken at different times) is to be recognized. The second column shows the segmented results when the learning process is stopped and the traffic sign has been recognized. Figure 4 demonstrates the learning behavior - a reduction in CPU time to recognize the traffic sign in one run of 12 images. Figure 5 shows the improvement in speed between the two schemes - scheme1 [26] and scheme2 [3]. Scheme 2 makes use of edge-border coincidence and global/local image segmentation to speed up the recognition process. Both schemes use the same learning algorithm.

Future Work: (a) Develop a complete reinforcement learning-based system for 3-D model-based ob-

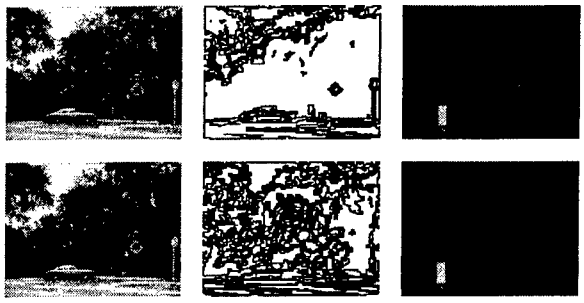


Figure 3: Integrated segmentation and matching results.

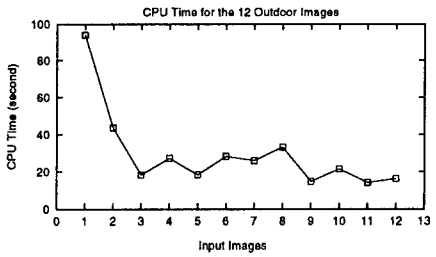


Figure 4: CPU time for one run of 12 outdoor images.

ject recognition with feedback among various levels. (b) Evaluate the performance of the technique for ATR application, (c) Learn algorithm parameters, develop algorithms and evaluation criteria for multisensor image segmentation and recognition, (d) Learn the optimal sensor combinations and cross-sensor validation of segmentation results.

3.1.2 Case-based reasoning for adaptive IU System

Problem: To automate acquisition of IU strategies, to integrate context with image properties, recognition algorithms and their parameters.

Approach: Most current model-based approaches to object recognition utilize geometric descriptions of object models, i.e., they emphasize the recognition problem as a characteristic of individual object models only. Various other factors, however, may influence the outcome of recognition in a real application such as photointerpretation. These factors include contextual information, sensor type, target type, scene models, and other non-image information. Using Case-Based Reasoning (CBR), successful recognition strategies (contextual information, algorithms, features, parameters, etc.) are stored in memory as cases and are used to solve new problems.

Since there are no algorithms that show acceptable performance over all different image sets that can be input to a system, we categorize images into classes and find the best algorithm for each class. When a new image is provided to recognize an object such as a particular aircraft type, the new image is first

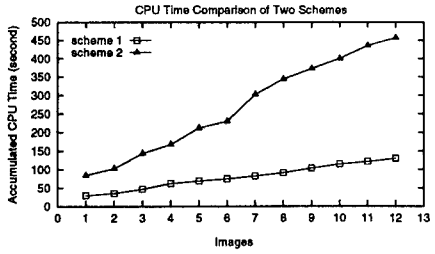


Figure 5: Comparison of accumulated CPU time for 5 different runs on 12 images.

categorized into the most similar class and then processed using the best algorithm known beforehand.

Categorization of images is, however, a very difficult problem. Instead of categorizing an image, a region of interest (ROI) is classified. For training images, ROIs are acquired and divided into classes by a human operator. The best algorithm is also selected by a human operator during training. Once images are categorized, characteristics of image sets are compiled statistically. These compiled probability distributions of values for each characteristic feature are utilized to find the most similar class. Characteristic features fall into two categories: contextual information and pure image metric information. Weather, time of image acquisition, and viewing angles are used as contextual information. Homogeneity factor, convexity factor, and agglomeration factor are suggested as pure image metrics information.

Accomplishments: We have developed the basic elements of the CBR paradigm. We have experimented extensively with a C-based algorithms for aircraft recognition in aerial photographs [19, 20, 21]. We have written code for characterizing image data sets.

Future Work: (a) Develop a prototype system which will have all the basic elements of CBR. (b) Select the best image metrics based on the discriminating power for categorizing images. (c) Develop reasoning, adaptation and indexing approaches that will make CBR an effective approach for IU applications [25].

3.2 Learning to Integrate Context with Clutter Models

Problem: To integrate contextual information with clutter models for target detection and recognition. Current image metrics commonly used to characterize images *do not* correlate well with the performance of target recognition systems.

Approach: The contextual parameters, which describe the environmental conditions for each training example, are used in a reinforcement learning paradigm to improve the clutter models and enhance target detection performance under multi-scenario situations [29]. New Gabor transform-based features and other statistical image features are used to capture the statistical properties of natural backgrounds in visible and FLIR images. The non-incremental self-organizing map approach commonly used in an unsupervised mode is extended, by the addition of a near-miss injection algorithm, and used as an incremental supervised learning process for clutter characterization [30].

Accomplishments: A fast algorithm to compute the Gabor transform of a given image has been implemented. We have implemented two new Gabor transform-based feature groups and tested their classification performance on natural backgrounds. Experimental results show that the two feature groups could capture certain characteristics of the backgrounds, which are consistent with our theoretical expectations based on the physical meaning of each attribute within the feature group.

Using 40 second generation FLIR images and four contextual parameters (time of the day, depression angle, range to the target and air temperature) and 5 feature groups, we find 100% detection rate, 10% false alarm rate and significant improvement in the confidence for classifying a feature cell (rectangular regions in an image) as a clutter or a target. The results have been compared. with and without contextual information [30].

Future Work: (a) Prove the convergence of the stochastic reinforcement learning algorithm for multi-feature cases. (b) Test the approach on a larger data set with a variety of contextual parameters. (c) Find the most influential environmental parameters for a given sensor, find how a feature group is affected by a given environmental parameter and find if we can make a feature invariant with respect to a given environmental parameter through normalization of the sensor data.

3.3 Learning for Input Adaptation and Feature Extraction

Problem: To improve the performance of an IU algorithm by adapting its input data to the desired form so that it is optimal for the given algorithm.

Approach: Two general methodologies for the performance improvement of an IU system are based on optimization of algorithm parameters and adaptation of the input. Unlike the genetic learning case for adaptive image segmentation, here we focus on the second methodology and use modified Hebbian learning rules to build adaptive feature extractors which transform the input data into the desired form for a given algorithm [35, 34]. Learning rules are based on different loss functions and are suitable for extracting expressive or discriminating features from the input.

Accomplishments: The feasibility of the approach is shown by designing an input adaptor for a thresholding algorithm for target detection in SAR, FLIR and color images. The results are excellent with input adaptor compared to the case with no input adaptor.

Future Work: (a) Develop transformations from input data to salient features needed for various classes of algorithms. (b) Compare performance with/without input adaptor for algorithms used in applications such as automatic target recognition.

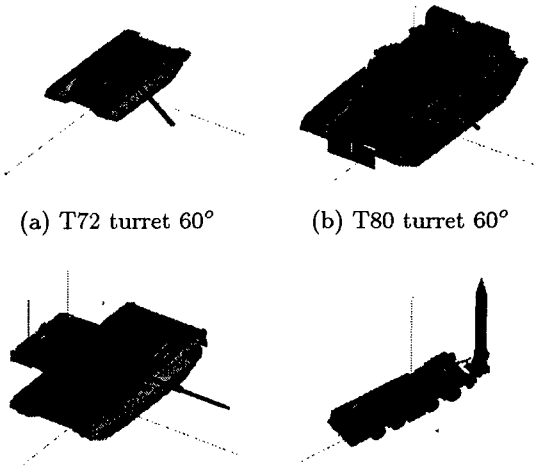
4 Automatic Target Detection and Recognition

The goals of our ATR research are to use sensor and geometric models and multiple representations (called physically-based modeling) for developing techniques for the recognition targets in multisensor imagery [6] and generic object recognition in complex aerial images. Our initial approach for indexing/matching in SAR images was based on using scattering centers and the Hausdorff distance measure [7, 32]. Since then we have focused on recognition of articulated and occluded objects and this approach is not suitable for it. We have made progress in the areas of recognition of articulated and occluded targets in SAR images using invariants and stochastic models [1, 16, 24]. We have also developed a Bayesian approach for the segmentation of SAR images and an approach for automatic model construction from inverse synthetic aperture radar images. Earlier we have developed and tested approaches based on Gabor wavelet representation [8] for (a) distortion-tolerant flexible matching for the recognition of occluded and nonoccluded targets in FLIR images, (b) computing salient structures in cluttered images, and (c) approaches for target detection in complex multimodal FLIR images.

4.1 Recognition of Targets in SAR Images

Problem: Develop techniques for indexing and matching to recognize articulated/occluded targets in SAR images.

Approach: Our invariants based approach is based on relative distances among the scattering centers to access a look-up table that generates the votes for the appropriate target and the azimuth. Using these results we can identify features which are on the turret and which are on the hull and can identify target, its body pose and the turret pose [1]. The power of the techniques is derived form the fact that it makes use of "azimuthal variance", both local and global constraints, high resolution data, "articulation invariants" and a voting mechanism as positive evidence for an efficient search [24].



(a) T72 turret 60° (b) T80 turret 60°

(c) M1a1 turret 90° (d) SCUD missile launcher

Figure 6: Articulated objects (not to scale).

Accomplishments: Figure 6 shows four sample targets which are used in our experiments. Using XPATCH generated data at 6in. resolution (10.0 GHz center frequency, 1.0 GHz bandwidth, 5.6° angular span), we have found that significant number of features do not typically persist over a few degrees of rotation. Averaging the results for 360 azimuths of the T72 tank, only about one-third of the 50 strongest scattering center locations remain unchanged for 1° azimuth (see Figure 7) and less than 5% persist for 10°. Figure 8 shows the articulation invariants. It shows the percentage of the strongest 50 scattering centers for the T72 tank that are in exactly the same location with the turret rotated 60° as they are with the turret straight forward. Figure 9 shows how the probability of correct identification varies with the percent invariance. Note that the recognition performance is excellent for invariance values greater than 40% (i.e., down to 60% spurious data). Recognition rate for varying amounts of occlusion (288,00 test cases) is shown in Figure 10. Note that it is consistent with the previous figure. Figure 11 compares (51,840 tests) the performance results of the articulated and occluded articulated targets for cases with the same number of valid scatterers. It shows the importance of relatively long distances and shows that object recognition approaches that combine both local and global constraints will be better than those which rely on local constraints only.

Future Work: (a) Test the approaches using real SAR data and quantify the performance, (b) Develop techniques for feature selection, (c) Develop matching techniques that account for complex feature types and 3D geometry [7], (d) optimize recognition performance with respect to feature extraction and feature types, (e) Develop a model for per-

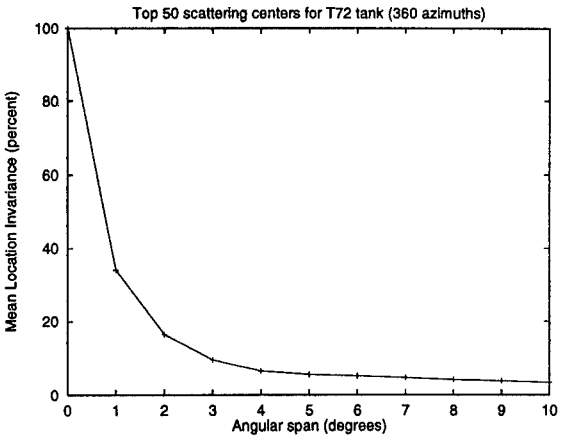


Figure 7: T72 azimuthal invariance.

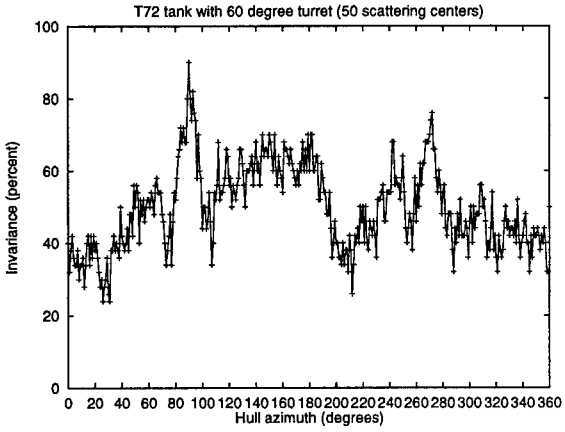


Figure 8: An example of Articulation invariants.

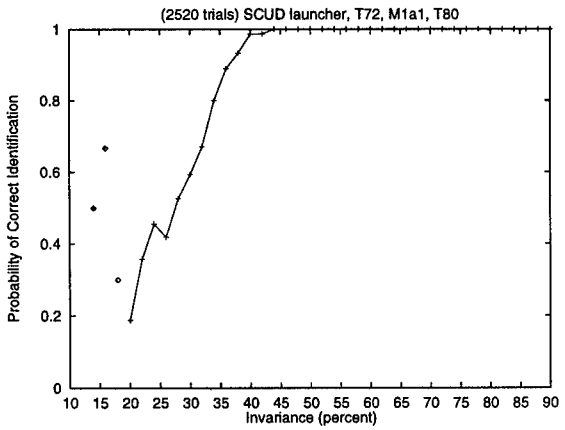


Figure 9: Recognition rate and articulation invariance (50 scatterers, average of 4 objects).

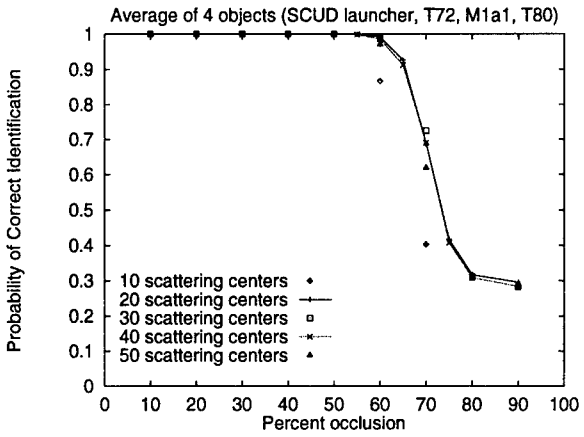


Figure 10: Recognition rate and occlusion percent.

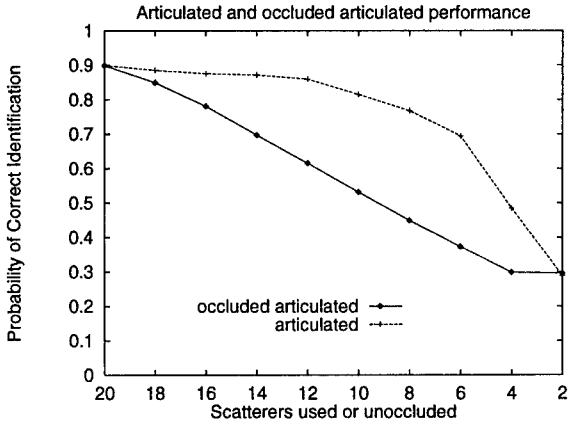


Figure 11: Articulated object and occluded articulated object performance results.

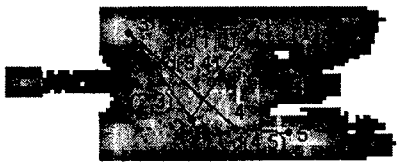


Figure 12: Example of an observation sequence superimposed on an image of T72 tank.

formance prediction.

4.2 Hidden Markov Modeling (HMM)-Based Approach for Indexing/Matching

Problem: Develop stochastic model-based techniques for indexing and matching to recognize articulated targets in SAR images.

Approach: The targets with pattern distortion caused by articulation and occlusion cannot be recognized by template matching. An alternative is to use a statistical method that can handle the possible configuration variations of the same object. Because of its stochastic nature HMM is suitable for describing patterns of variation. The key elements of HMM include: finding the probability of the observations given the model, finding the most likely state trajectory given the model and observation, and adjusting the parameters of HMM to model the observation sequence better. The basic idea is imagining the features as emitting patterns of some hidden statistical model. We can sort available features to get appropriate sequences as the observation sequences of HMM. These sequences will represent the particular pattern of point features. It is reasonable to use the relative locations and relative magnitudes of these point features to obtain observation sequences (see Figure 12). Sequence based on relative amplitudes of SAR image is O1 = Mag 1, Mag 2, Mag 3, ..., Mag n. Selected sequences based on geometrical relationship are: O2 = d(1,2), d(2,3), d(3,4), ..., d(n-1,n), d(n,1), O3 = d(1,2), d(1,3), ..., d(1,n), O4 = d(2,1), d(2,3), ..., d(2,n), O5 = d(3,1), d(3,2), ..., d(3,n).

Accomplishments: We have used HMM for recognition of occluded objects in XPATCH generated data as described above. Examples of occlusion in training and test cases are shown in Figure 13. During training we find the optimal number of symbols (4) and states (5) for HMM. Using 325,000 training samples (5-10% occlusion) and 81,000 testing samples (5-50%occlusion) for 5 classes we find the results as shown in Figure 14. The results obtained from individual models are combined by an algorithm to achieve the results shown in this figure. The dotted lines show the worst and the best performance that was achieved with 5 HMM models (O1 to O5).

Future Work: (a) Test the approach for articulation, and occlusion with articulation, (b) Test the approach on real SAR data, (c) Develop techniques to find the optimal number of HMM models for various targets,(d) Develop methods to integrate results from different HMM models,

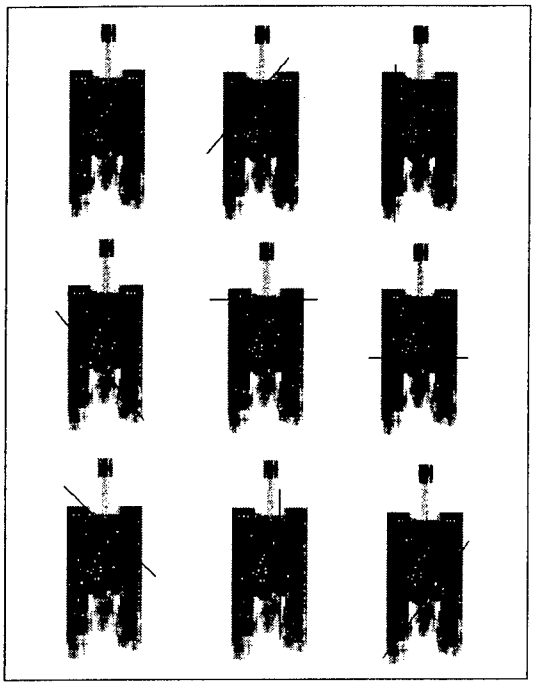


Figure 13: Scattering centers of T72 tank at azimuth 0°, part of scattering centers are occluded from a particular direction (0-8).

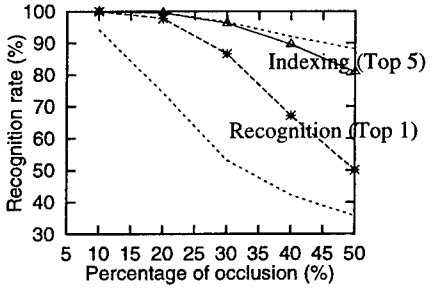


Figure 14: Performance of integrated models: using integrated models O1 to O5. The results for recognition (Top 1) and indexing (Top 5) candidates are superimposed on the figure shown in (a).

4.3 Other SAR related work

We have developed a Bayesian approach [22] using dynamically selected neighborhoods for the segmentation of SAR images. The approach allows a variety of a prior information to be explicitly included in the image segmentation task. Preliminary results have been shown on simulated data.

We have also constructed the models automatically for object recognition using ISAR images. Given a set of ISAR data of an object of interest, structural

features are extracted from the images. Statistical analysis and geometrical reasoning are then used to analyze the features to find spatial and statistical invariance so that a structural model of the object suitable for object recognition can be constructed. A novel feature of the approach is that it uses the persistency of scattering centers computed during training phase to extract good scattering centers during testing phase. Four objects (Camaro, Dodge Van, Dodge Pickup and Bulldozer) are used to demonstrate the results. There are 351 images in both the training and the testing data for each object. The testing data are offset by 0.2 degree in azimuth from the corresponding training data. The results are significantly better compared to the case when we do not use the persistency of scattering centers during on-line phase. The results show that this approach is promising for automatic model construction [33].

4.4 Gabor Wavelets for Target Recognition and Target Detection

Using Gabor wavelets representation we have developed model-based, distortion-tolerant flexible matching techniques for recognition of occluded and nonoccluded targets under varying environmental conditions [31]. The key idea is to use magnitude, phase and frequency measures of Gabor wavelet representation in an innovative flexible matching approach that can provide robust recognition. The Gabor grid, a topology-preserving map, efficiently encodes both signal energy and structural information of an object in a sparse multi-resolution representation. Flexible matching between the model and the image minimizes a cost function based on local similarity nd geometric distortion of the Gabor grid. Grid erosion and repairing is performed whenever a collapsed grid, due to object occlusion, is detected. We have performed a variety of experiments with second generation FLIR data and synthetic targets exhibiting varying signatures with changing environmental conditions. The results are reported in [31].

We have developed a new feature ("composite phase") based on Gabor wavelets. Also we have developed techniques for the computation of salient structures and target detection using wavelets [2].

5 Other Research

Other areas of ongoing work include navigation and obstacle detection [4, 13, 15]. We are developing a mobile testbed, called *UCRover* for experiments in perception and learning. We have developed model-based generic object recognition approaches for qualitative recognition of aircraft in perspective aerial imagery and tested them on complex aerial images [19, 20, 21]. We have also done research on terrain interpretation using multispectral images [14].

6 Conclusions

We have developed promising approaches and obtained good results to solve some of the fundamental problems in IU that will have strong impact in solving real-world applications. In the coming years our focus will be the development of new algorithms and the end-to-end complete system that integrates recognition, learning and image databases for image exploitation using SAR, visible and multispectral imagery. We shall emphasize the performance evaluation of our algorithms and systems to measure improvements over current approaches.

References

[1] J. Ahn and B. Bhanu. Matching of objects with articulation and occlusion in SAR images. In *Proc. ARPA Image Understanding Workshop*, New Orleans, LA, 1997. May 13-15.

[2] B. Bhanu. Image understanding research at UC Riverside: Robust recognition of objects in real-world scenes. In *Proc. ARPA Image Understanding Workshop*, pages 117–128, Palm Springs, CA, 1996. February 13-15.

[3] B. Bhanu, X. Bao, and J. Peng. Reinforcement learning integrated image segmentation and object recognition. In *Proc. ARPA Image Understanding Workshop*, New Orleans, LA, 1997. May 13-15.

[4] B. Bhanu, W. Burger, and R.N. Braithwaite. Perception for outdoor navigation. Submitted to *IEEE Trans. on Robotics and Automation*, 1996 (under revision).

[5] B. Bhanu and S. Das. *Computational Learning for Adaptive Computer Vision.* Plenum Publishing Company, 1997 (forthcoming).

[6] B. Bhanu, D.E. Dudgeon, E.G. Zelnio, A. Rosenfeld, D. Casasent, and I.S. Reed. Introduction to the special issue on automatic target detection and recognition. *IEEE Trans. Image Processing*, 6, no. 1:1–6, January 1997.

[7] B. Bhanu, G. Jones, J. Ahn, M. Li, and J. Yi. Recognition of articulated objects in SAR images. In *Proc. ARPA Image Understanding Workshop*, pages 1237–1250, Palm Springs, CA, 1996. February 13-15.

[8] B. Bhanu, G. Jones, J. Yi, J. Ahn, S. Zhang, M. Li, T. Ferryman, and B. Tian. Gabor wavelets for automatic target detection and recognition. Technical report, UC Riverside, July 1996. (Report to DARPA).

[9] B. Bhanu, S. Lee, and S. Das. Adaptive image segmentation using genetic and hybrid search methods. *IEEE Trans. Aerospace and Electronic Systems*, 31, no. 4:1268–1291, October 1995.

[10] B. Bhanu, S. Lee, and J. Ming. Adaptive image segmentation using a genetic algorithm. *IEEE Trans. Systems, Man, and Cybernetics*, 25, no. 12:1543–1567, December 1995.

[11] B. Bhanu and J. Ming. Recognition of occluded objects: A cluster-structure algorithm. *Pattern Recognition*, 20, no. 2:199–211, 1987.

[12] B. Bhanu, J. Peng, Y.-J. Zheng, S. Rong, and X. Bao. Multistrategy learning for computer vision. Technical report, UC Riverside, March 1996. (Technical Report to AFOSR and DARPA).

[13] B. Bhanu, B. Roberts, D. Duncan, and S. Das. A system for obstacle detection during rotorcraft low-altitude flight. *IEEE Trans. Aerospace and Electronic Systems*, 32, no. 2:875–897, April 1996.

[14] B. Bhanu, P. Symosek, and S. Das. Analysis of terrain using multispectral images. *Pattern Recognition*, 30, no. 2, February 1997.

[15] B. Bhanu, P. Symosek, S. Snyder, B. Roberts, and S. Das. Synergism of binocular and motion stereo for passive ranging. *IEEE Trans. Aerospace and Electronic Systems*, 30, no. 3:709–721, July 1994.

[16] B. Bhanu and B. Tian. Multiple stochastc models for recognition of objects in SAR images. In *Proc. ARPA Image Understanding Workshop*, New Orleans, LA, 1997. May 13-15.

[17] B. Bhanu, X. Wu, and S. Lee. Genetic algorithms for adaptive image segmentation, pages 269–298. *Early Visual Learning*, edited by S.K. Nayar and T. Poggio. Oxford University Press, March 1996.

[18] S. Das and B. Bhanu. Computational vision: A learning perspective. Submitted to *ACM Computing Surveys*, 1996 (under revision).

[19] S. Das, B. Bhanu, and C.C. Ho. Multiple representations for generic object recognition. *Image and Vision Computing*, 14:323–338, 1996.

[20] S. Das, B. Bhanu, X. Wu, and R.N. Braithwaite. Model-based qualitative object recognition. Submitted to *Pattern Recognition*, 1995 (revised).

[21] S. Das, B. Bhanu, X. Wu, and R.N. Braithwaite. Qualitative Recognition of Aircraft in Perspective Aerial Imagery. Chapter in *Advances in Image Processing and Machine Vision*, edited by J. Sanz. Springer Verlag, 1996.

[22] T. A. Ferryman and B. Bhanu. A Bayesian approach for the segmentation of SAR images using dynamically selected neighborhoods. In *Proc. ARPA Image Understanding Workshop*, pages 891–896, Palm Springs, CA, 1996. February 13-15.

[23] A. Gupta. Visual information retrieval technology, a virage perspective. Technical report, Virage Inc., 1995.

[24] G. Jones and B. Bhanu. Invariant features for the recognition of articulated and occluded objects in SAR images. In *Proc. ARPA Image Understanding Workshop*, New Orleans, LA, 1997. May 13-15.

[25] J. Ming and B. Bhanu. A multistrategy learning approach for object recognition. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 1997 (in press).

[26] J. Peng and B. Bhanu. Closed-loop object recognition using reinforcement learning. Submitted to *IEEE Trans. Pattern Analysis and Machine Intelligence*, Feb. 1997 (revised), An earlier version in CVPR'96.

[27] J. Peng and B. Bhanu. Delayed reinforcement learning for closed-loop object recognition. Submitted to *IEEE Trans. Systems, Man and Cybernetics*, March. 1997 (revised), An earlier version in ICPR'96.

[28] S.J. Press and K. Shigemasu. *Contributions to Probability and Statistics: Essays in Honor of Ingram Olkin*, chapter Bayesian Inference in Factor Analysis, Chapter 15, L.J. Gleser and M.D. Perlman and S.J. Press and A.R. Sampson (editors), pages 271–287. 1989.

[29] S. Rong and B. Bhanu. Modeling clutter and context for target detection in infrared images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 106–113, San Francisco, CA, 1996. June 16-20.

[30] S. Rong and B. Bhanu. Context reinforced background model for target detection. Submitted for journal publication, 1997.

[31] X. Wu and B. Bhanu. Gabor wavelet representation for 3-D object recognition. *IEEE Trans. Image Processing, Special Issue on Automatic Target Recognition*, 6, no. 1:47–64, January 1997.

[32] J.H. Yi, B. Bhanu, and M. Li. Target indexing in SAR images using scattering centers and Hausdorff distance. *Pattern Recognition Letters*, 17:1191–1198, September 1996.

[33] S. Zhang and B. Bhanu. Automatic model construction for object recognition using ISAR images. In *Proc. 13th Int. Conf. on Pattern Recognition*, volume IV, pages 169–173, Vienna, Austria, 1996. August 25-30.

[34] Y.-J. Zheng and B. Bhanu. Adaptive object detection using modified Hebbian learning. In *Proc. 13th Int. Conf. on Pattern Recognition*, volume IV, pages 164–168, Vienna, Austria, 1996. August 25-30.

[35] Y.-J. Zheng and B. Bhanu. Adaptive object detection from multisensor data. In *Proc. IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pages 633–640, Washington, DC, December 1996, Submitted to IEEE Trans. SMC.

Image Browsing and Retrieval Research at Stanford*

Leonidas J. Guibas and Carlo Tomasi
Computer Science Department, Stanford University
Stanford, CA 94305
[guibas,tomasi]@cs.stanford.edu
http://robotics.stanford.edu/users/[guibas,tomasi]/bio.html

Abstract

In the past year, work on image browsing and retrieval at Stanford has concentrated on the development and use of consistent representations of color, shape, and texture content in images. These compact representations, called *signatures*, are more flexible than feature vectors or histograms, because they have variable length and imply no ordering among primitives and no quantization. A perceptually useful metric for color signatures has been defined based on what we call the "earth mover's distance." A new technique for arranging pictures so that similar images are grouped together has been developed, based on multi-dimensional scaling. In color-based retrieval, a new Netscape-based interface yields access to a 5,000 image database. Retrieval is fast, and display of the results is made more intuitive by our two-dimensional layouts. An image navigator provides an entirely new paradigm for interaction with a picture database. Based on multi-dimensional scaling, the navigator arranges all (or a large sample of) the images in the database into a three-dimensional space, and the user can quickly home in to the regions of interest, and form a mental model of what is in the database.

1 Introduction

The literature on image retrieval is growing, with several efforts in both academia [Guibas and Tomasi, 1996, Forsyth *et al.*, 1996, Pentland *et al.*, 1996, Stricker, 1996, Santini and Jain, 1996, Wan and Kuo, 1996, Ravela *et al.*, 1996, Swain and Ballard, 1991, Jacobs *et al.*, 1995] and industry [Faloutsos *et al.*, 1994, Virage, 1997, Gallant and Johnston, 1995]. The main thrust of our work is the definition of basic image representations that are most appropriate for image search. With the aim of a unified treatment, we have developed the notion of a *signature* to summarize image appearance. Signatures can represent the color, shape, or texture content of an image. They are more flexible than feature vectors and histograms, as they imply no fixed number or ordering of feature primitives, as in vectors, nor fixed-pitch quantization of feature values, as in histograms. Color and shape signatures are described in sections 2 and 7. Texture signatures are one of the main goals of our current research.

By using a single representation format for the three different modalities considered in our work, that is, color, shape, and texture, we hope to make our retrieval mechanisms essentially uniform across modalities. This should lead not only to efficiency and simplicity, but also to conceptual consistency. We believe that it will be easier to combine searches in these different modalities if the underlying representations are mutually consistent.

The other main ingredient of a retrieval system, besides signatures, is a perceptually meaningful measure of similarity between two images. We have defined such a measure based on what we call the "Earth Mover's Distance" (section 3). With these two ingredients, the pictures in a database can be organized so as to keep similar images close to each other. We are currently working on organizing these image signatures into efficient data structures for sublinear nearest-neighbor retrieval. In addition, a similarity metric between images leads to methods for laying out either all the images in the database, or a sample thereof, or a small number of mutually related images, and for displaying these images in an intuitive way for the user. The mathematical tool we used for the creation

*This work was sponsored by the Defense Advanced Research Projects Agency under contract DAAH04-94-G-0284 monitored by the US Army Research Office. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the United States Government, or Stanford University.

of this layout is multi-dimensional scaling (section 4).

In our work on color retrieval, we built a Netscape-based interface that allows fast retrieval from a 5,000 image database based on color signatures (section 5). Furthermore, we demonstrated the notion of a database navigator, in which many of the images in a database are laid out in three-dimensional space (section 6). The user then navigates in this space with a joystick. The main advantage of this new interaction paradigm is that the content of the database is conveyed to the user all at once, rather than piecemeal, as in the more standard query/response protocol. A global view lets the user form a mental picture of the database, just as one forms a mental picture of the contents of, say, a bookstore by browsing in it for some time (see figure 1). If the images are arranged in a coherent fashion, consistent with our similarity metric, the ordering rationale is easily learned by the user without being explicitly identified. At a more local level, again thanks to our metric, the small number of images returned in response to a query in the more traditional query/response operating mode can be displayed so as to emphasize similarities and differences among the images.

In shape-based retrieval, we continued our earlier work on developing shape indices by recording "what basic shape appears where in the image." We successfully experimented with a data-base of scanned-in Chinese characters (figure 2) and used geometric hashing to make our indices invariant under a transformation group. We are currently experimenting with some variants of shape signatures and definitions of the earth mover's distance for them.

In the following sections, we outline the main aspects of our color- and shape-based retrieval work. In section 8, we discuss research plans for next year and beyond. More details on the technical aspects of our research are given in two companion papers in these proceedings[Rubner *et al.*, 1997, Cohen and Guibas, 1997b].

2 Color Signatures

The color information of each image is reduced to a compact representation that we call the *signature* of the image. In general a signature contains a varying number of points in a Euclidean space where a weight is attached to each point. In the case of color images, the points represent clusters of similar colors in CIE-LAB space, and the weight of a point is the fraction of the image area with that color. The signatures thus obtained are compact: the color distribution of an entire image is summarized by a handful of points, typically eight to twelve. Since signatures represent distributions in the CIE-LAB color space, they are perceptually significant, in that Euclidean distances between points are strongly correlated with perceptual differences. Because of clustering, small variations in the colors of an image have little effect on signatures, thereby providing a moderate degree of invariance to changes of viewpoint and lighting. Finally, signatures are simple and flexible abstractions. In fact, the cloud of weighted points that makes up a color signature lives in the low-dimensional space of colors. Furthermore, just as objects and concepts are described in English by sentences with a variable number of words, so images are summarized by a variable number of colors in a signature. The ordering of colors in not meaningful, and is therefore not used. The relative importance of the various colors is explicitly represented by the weight of each signature component, and is therefore immune from the quantization problems inherent in color histograms.

3 The Earth Mover's Distance

We define the distance between two signatures to be the minimum amount of 'work' needed to transform one signature into the other. The work needed to move a point, or a fraction of a point, to a new location is the portion of the weight being moved, multiplied by the Euclidean distance between the old and the new locations. When changing one signature to another, the work is the sum of the work done by moving the weights of the individual points of the source signature to those of the destination signature. We allow the weight of a single source signature point to be partitioned among several destination signature points, and vice versa. The distance between the source and destination signatures is then defined to be the minimum amount of work necessary to thus move the weight of the source to that of the destination signature. We call this distance function the *earth mover's distance.*

Computing the earth mover's distance can be formulated as a linear programming (LP) problem [Rubner *et al.*, 1997]. Given the compact nature of color signatures, this LP problem is relatively small. Still, since computing this distance is the main operation in our image retrieval systems, we are devoting considerable efforts to making this solution as fast as possible. Currently, the distance between two im-

ages is computed in a small fraction of a second. Bounds can be used both to exclude from consideration images that are too distant from the query and to abort computation of a distance once it is certain to exceed a certain value. These refinements are briefly discussed in section 8.

4 Multi-Dimensional Scaling

Our earth mover's distance quantifies the perceptual difference that separates two signatures. Consequently, each signature can be represented by a single point in a suitably high-dimensional space, such that distances between these points are equal to the earth mover's distances between the corresponding signatures. The computation of the coordinates of these high-dimensional points is called an *embedding*. However, humans can only visualize low-dimensional spaces, typically in two or three dimensions. We then look for an approximate embedding, rather than for an exact one.

The approximate embedding problem was formalized by Kruskal [Kruskal, 1964] into the so-called Multi-Dimensional Scaling (MDS) problem. Using MDS can assist navigation in the space of images both locally and globally, as we now illustrate.

5 Fast Color-Based Retrieval

Our Netscape-based retrieval system lets the user find images based on their color distributions. Queries can either use or ignore positional information. Currently, position is handled by splitting up a query window into an array of 5 by 5 rectangles, each of which can be searched independently. Queries are either specified by coloring each rectangle with a different color, or by using another image, returned by the retrieval system, as a query. We found it useful to also provide a "random query" button, which returns random images. These can be used as starting points for a more focused query.

Performing MDS on the images returned from a query gives us a better way to display the query results. Instead of the traditional one-dimensional list of images sorted by their distances from the query, we can display a two or three dimensional map of the images, where each image is positioned according to the MDS result. In this way we are presenting information reflecting $\binom{n}{2}$ distances, instead of only n in the traditional method. In addition to visually representing the relative distances between *all* pairs of images, images with similar color content will group together.

6 Navigating in a Space of Color Images

Performing MDS on a large set of images can help the user understand the space of color images at whole. In the resulting displays, discussed in [Rubner *et al.*, 1997], images end up grouping by a combination of their dominant chroma, lightness, and saturation. We emphasize that these criteria are "discovered" by multi-dimensional scaling (section 4), not hard-coded by the programmer. As a consequence, higher dimensional MDS can be done on the image database where different characteristics of the images will be revealed. These higher-dimensional layouts can be displayed through different projections onto two- or three-dimensional spaces. Given one of these layouts, including the simple three-dimensional ones we have produced in our recent work, when the user looks for a sunset she sees immediately where to go. At a glance, she can write off most of the data-base, and home in to the "sunset-looking" part of it. At the same time, the user forms a mental picture of the entire data base. Everything is seen in coarse detail, and the impression arises of grasping the overall data-base content, at least in terms of color distributions. Given a joystick that lets the user get closer to the area of interest, the system conveys at the same time focus, because nearby images are large on the display, and context, because all or most other images are still visible at a distance. As the user moves about, she has the comforting impression that the whole data-base is there all the time, rather than being handed down to her in small disconnected fragments.

7 Shape Signatures

We base our shape signatures on image edges. After running an edge detector on the image, we link edgels into chains, and then fit a number of primitive geometric shapes to the resulting chains. Our initial implementation is based on fitting line segments to these chains. We also intend to explore fitting with other simple primitives, such as corners, circular arcs, S-shapes, and so on. These are the kinds of primitives which the user might quickly sketch with a drawing program as a way of indicating the shape content of the image.

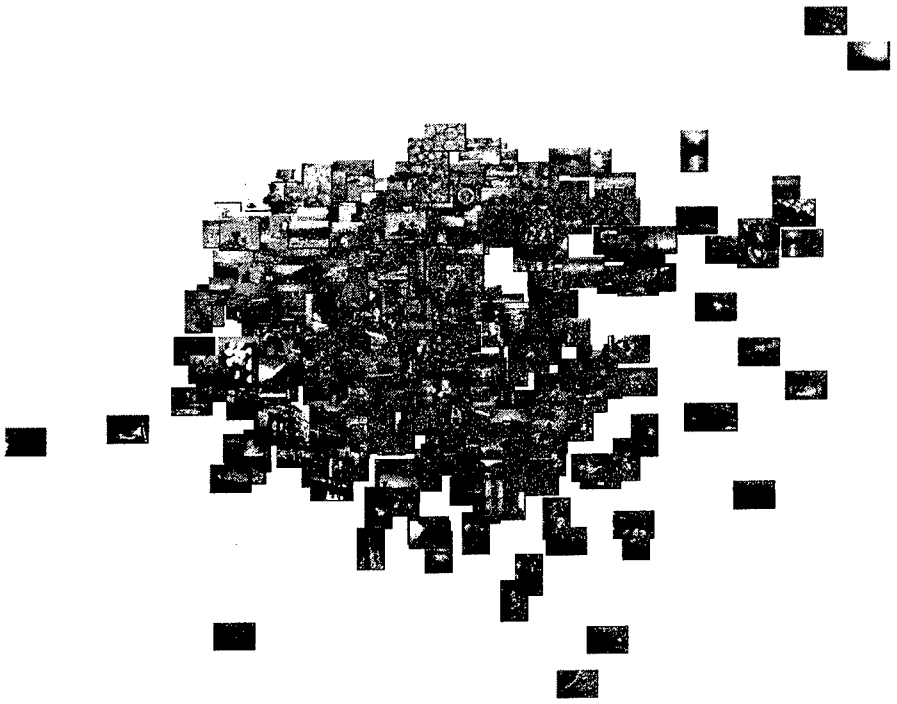


Figure 1: 2D MDS map of 500 images. A color version can be viewed at `http://vision.stanford.edu/irs/colorpics.html`.

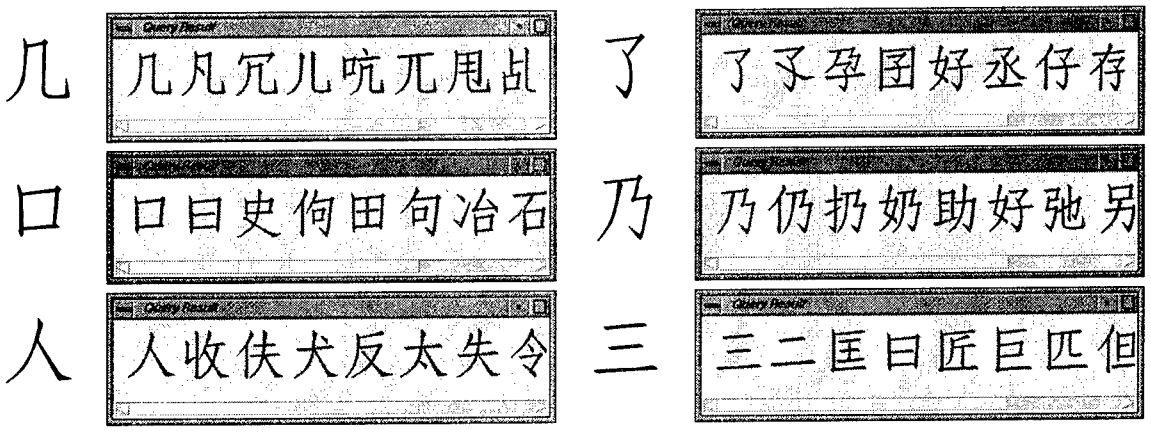


Figure 2: Sample queries into our chinese character database, with corresponding results. Each query takes about one second on an SGI Indy.

The nature of a shape signature element is then to record a good fit of one of these basic primitives with the edges found in the image. Just as in the case of color, what we store describes 'what is where' in terms of significant shape features. Thus shape signature elements come in different flavors, depending on the basic shape that has been matched. Note that in general there will be many good matches of a given basic shape into an image and our goal is to choose a small and representative subset. For example, near any very good match there will be other (less) good matches that are dominated by it. Clearly we do not want to record the latter in the signature. We have recently had some progress in dealing successfully with this issue [Cohen and Guibas, 1997a].

We have experimented extensively with such shape descriptors and obtained very good results for a data-base of illustrations from a geometry text, where of course there is no noise in the shape data [Cohen and Guibas, 1996]. But we also obtained good results with a data-base of scanned-in Chinese character shapes [Cohen and Guibas, 1997b].

Shape signature elements correspond to transformations mapping a basic shape primitive into its location in the image. As such they can also be viewed as points in an appropriate low-dimensional parameter space; in this case the weight can indicate the size or length of the shape primitive. However, the earth mover's distance must be modified to be used successfully with such shape signatures. The reason is because, for example, the same edge in two similar images might be indexed as one long segment in one image, but as two shorter segments in the other due to noise, etc. We are currently experimenting with various adaptations of the earth mover's distance for matching "fragments to a whole."

8 Future Work

The concepts we have developed in our recent work form a consistent, solid basis for the construction of retrieval systems based on color, shape, and texture. At the same time, the prototype systems we have built to demonstrate their effectiveness, although efficient and non-trivial in size, are merely proofs of concept. Several promising avenues of research present themselves.

One set of problems relates to the fine tuning and improvement of our existing demonstration systems. The interface must be made more flexible, and allow mixing of image details and colors from the given palette. The MDS display of the returned images must be made "alive" by letting the user click on it to provide a component for the next query.

The core computation of the earth mover's distance must be made as efficient as possible. We have discovered that the dual LP problem is in our case much faster than the primal. Also, the distance between two signatures can be shown to be bounded from below by the distance between their centers of mass. This can lead to dramatic shortcuts in the computation. Speed-ups can be achieved also by using the special structure of the problem internally to the LP computation, by using an interior-point algorithm, rather than a simplex method, and by terminating the computation of a distance from the current query when this distance is provably too large to be of interest.

Another set of issues arises from the application of the signature concept to the description of texture. Although we have already developed the main elements of texture representation and analysis [Rubner and Tomasi, 1996], texture signatures and the meaning of the earth mover's distance for texture are still to be explored. We must also determine the significance of MDS in texture space. Similarly, much remains to be explored in applying these ideas to shape signatures, as discussed above.

Perhaps the hardest problems relate to the combination of different query modalities. How can we incorporate spatial position in greater detail than by our device of a 5 by 5 partition of the query window? How can we more meaningfully combine the 25 results from querying each of the windows? And how can we combine query by color, shape, and texture in order to give the user a flexible retrieval tool? We plan to pursue these questions in our next year of research and beyond.

Acknowledgments

The authors wish to thank Tamara Munzner for building the graphics software used in the color-space navigator.

References

[Cohen and Guibas, 1996] S. Cohen and L. J. Guibas. Shape-based illustration indexing and retrieval: some first steps. In *ARPA*

Image Understanding Workshop, pp. 101-108, 1996.

[Cohen and Guibas, 1997a] S. Cohen and L. J. Guibas. Partial matching of planar polylines under similarity transformations. In *Proc. 8th ACM-SIAM Symp. Discrete Algorithms*, pp. 777–786, 1997.

[Cohen and Guibas, 1997b] S. Cohen and L. J. Guibas. Shape-based image retrieval using geometric hashing. In *Proceedings of the ARPA Image Understanding Workshop*, New Orleans, LA, 1997.

[Faloutsos *et al.*, 1994] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, 1994.

[Forsyth *et al.*, 1996] D. Forsyth, J. Malik, M. Fleck, H. Greenspan, and T. Leung. Finding pictures of objects in large collections of images. In *International Workshop on Object Recognition for Computer Vision*, Cambridge, UK, April 1996.

[Gallant and Johnston, 1995] S. I. Gallant and M. F. Johnston. Image retrieval using image context vectors: first results. In *IS&T/SPIE Symposium on Electronic Imaging: Science and Technology*, volume 2420, pages 82–94, San Josè, CA, February 1995.

[Guibas and Tomasi, 1996] L. J. Guibas and C. Tomasi. Image retrieval and robot vision research at Stanford. In *Proceedings of the ARPA Image Understanding Workshop*, pages 101–108, Palm Springs, CA, 1996.

[Jacobs *et al.*, 1995] C. Jacobs, A. Finkelstein, and D. Salesin. Fast multiresolution image querying, *Proc. ACM SIGGRAPH'95*, pp. 277–286, 1995.

[Kruskal, 1964] J. B. Kruskal. Multi-dimensional scaling by optimizing goodness-of-fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.

[Pentland *et al.*, 1996] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, June 1996.

[Ravela *et al.*, 1996] S. Ravela, R. Manmatha, and E. M. Riseman. Image retrieval using scale space matching. In B. Buxton and R. Cipolla, editors, *Computer Vision – ECCV96*, pages 273–282, Cambridge, UK, April 1996. Springer-Verlag.

[Rubner and Tomasi, 1996] Y. Rubner and C. Tomasi. Coalescing texture descriptors. In *Proceedings of the ARPA Image Understanding Workshop*, pages 927–935, Palm Springs, CA, 1996.

[Rubner *et al.*, 1997] Y. Rubner, L. J. Guibas, and C. Tomasi. The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. In *Proceedings of the ARPA Image Understanding Workshop*, New Orleans, LA, 1997.

[Santini and Jain, 1996] S. Santini and R. Jain. Similarity queries in image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR96)*, pages 646–651, San Francisco, CA, 1996.

[Stricker, 1996] M. A. Stricker. Color indexing with weak spatial constraints. In I. K. Sethi and R. C. Jain, editors, *Photonics West '96 — Storage and Retrieval for Still Image and Video Databases IV*, pages 2670–04, San Josè, CA, February 1996.

[Swain and Ballard, 1991] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[Virage, 1997] `http://www.virage.com/`.

[Wan and Kuo, 1996] X. Wan and C. J. Kuo. Image retrieval with multiresolution color space quantization. In *Proceedings of the SPIE*, volume 2898, pages 148–159, 1996.

IMAGE EXPLOITATION
(IMEX)
TECHNICAL PAPERS

Thermal Invariants for Material Labeling and Site Monitoring Using Midwave Infrared Imagery: Initial Results

C. Stewart
Rensselaer Poly. Inst.
Troy, NY 12180

V. Snell D. Hamilton
CMA Consulting
Schenectady, NY 12301

J. Mundy*
GE Research
Schenectady, NY 12301

Abstract

In ongoing work we are exploring the use of four different thermal invariants to identify materials and to detect changes in materials in sequences of thermal images. An user-selected set of regions are located in each thermal image through the use of a detailed site model, which we have built as part of this work. Invariants are calculated in each image from the temperature values in the regions and from their hypothesized material types. These invariant values are compared across the sequence to determine their stability and accuracy in discriminating materials.

1 Introduction

This paper describes preliminary work studying the use of thermal invariants to identify materials and to monitor changes in materials using sequences of thermal (infrared) image data. These materials may be the exterior faces and roofs of buildings, the surfaces of roads, airstrips or parking lots, or the external surfaces of objects outside buildings. The work is part of a joint project between the GE Center for Research and Development and Wright Labs.

A thermal invariant [Arnold *et al.*, 1996, Michel *et al.*, 1997, Nandhakumar and Velten, 1994, Nandhakumar *et al.*, 1996] is a function mapping several quantities to a single real number; these quantities include surface temperatures calculated from thermal imagery, known thermophysical properties of hypothesized surface materials and, potentially, ambient temperature measurements. In principle, when surface and ambient temperatures are measured accurately and when the correct materials are hypothesized, this number should be stable with respect to time, regardless of the imaging conditions. Furthermore, the number should ideally be the same for different sets of surfaces having the same material properties. Therefore, thermal invariants can potentially be used to identify surface material types and to monitor changes in surface properties.

Our work, which focuses on the application of thermal invariants rather than the theory, differs from past work in several ways. First, most previous work has applied thermal invariants to object recognition rather than material identification and change monitoring. This allows us to base our invariant calculations on large image regions rather than on individual pixels. Second, our test data is "Midwave Infrared" (MWIR) imagery rather the "Longwave Infrared" (LWIR) imagery used in previous studies of thermal invariants. MWIR sensors measure radiation in the 3-5 μm wavelength range instead of the 8-14 μm range of LWIR

*This work was supported by DARPA contract F33615-94-C-1529, monitored by Wright Patterson Airforce Base, Dayton, OH. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the United States Government, Rensselaer Polytechnic Institute, or General Electric. This work was done while the first author was on sabbatical at GE.

range. The consequences of this for the physical model on which the thermal invariants are based have not yet been explored. Third, we only consider imagery taken at night since other imaging modalities will be used for daylight hours. This simplifies the calculations since the effects of solar radiation can be ignored.

A crucial requirement in developing and studying thermal invariants is the ability to locate the same region in multiple images. In our study this is done using both a detailed site model and the intrinsic and extrinsic parameters of a perspective camera model for each image. Hence, a region in any one image may be backprojected onto the site model and projected onto any other image. This allows regions to be tracked across a time sequence of thermal images. We have built a detailed site model using visible light intensity images and a CAD model from Wright-Patterson Air Force Base. The thermal data were all acquired there.

The remainder of this paper is organized into four main sections. Section 2 summarizes the thermal invariants studied thus far. It outlines the physical model on which they are based and sketches the method for deriving the invariants. Section 3 describes techniques for using these thermal invariants to identify and monitor materials. Section 4 presents a set of initial results. Section 5 concludes the paper by outlining ongoing work.

2 Thermal Invariants

The first thermal invariant proposed in the literature is based on an argument about the thermal history of object surfaces [Gauder *et al.*, 1993]. All other suggested thermal invariants are based on a model of the thermal equilibrium of an imaged surface. This model [Nandhakumar and Aggarwal, 1988] is

$$W_{abs} = W_{rad} + W_{cv} + W_{st} + W_{cnd} \quad (1)$$

Each of the W terms is a heat flux. The left side of the equality is the energy absorbed from the environment at the surface. The right is the energy lost from the surface.

W_{abs} is modeled as being exclusively from solar energy and is written as

$$W_{abs} = W_I \cos\theta_I \alpha_s \quad (2)$$

where W_I is the incident solar radiation at the surface element, $\cos\theta_I$ is the cosine of the angle between the sun and the object surface normal, and α_s is the solar absorptivity of the surface. At night, $W_{abs} = 0$.

W_{rad} is the heat loss due to radiation from the surface:

$$W_{rad} = \epsilon\sigma(T_s^4 - T_{amb}^4) \quad (3)$$

where ϵ is the surface emissivity, σ is the Stefan-Boltzmann constant, T_s is the measured surface temperature in (degrees) Kelvin, and T_{amb} is the ambient temperature, also in (degrees) Kelvin.

W_{cv} is the heat loss due to convection:

$$W_{cv} = h(T_s - T_{amb}) \quad (4)$$

where h is the "average convection heat transfer coefficient" which depends on a variety of environmental and surface factors (see discussion in [Nandhakumar and Aggarwal, 1988]).

W_{st} is the stored energy at the object surface:

$$W_{st} = C_T \frac{dT_s}{dt} \quad (5)$$

where C_T is the lumped thermal capacitance of the object.

W_{cnd} is the heat conducted to the object interior:

$$W_{cnd} = -k \frac{dT_s}{dx} \quad (6)$$

where k is the thermal conductivity of the material and x is the distance below the surface.

2.1 Deriving Thermal Invariants

The above equations include a variety of parameters. Some of these are known constants. Some, such as the ambient temperature and the surface temperature, may be measured from the environment. Some are known when hypotheses about materials are specified. The rest, which are often called the "driving conditions", are

difficult to measure or hypothesize. To make use of equation 1, these parameters must be eliminated from consideration. This is done by using measurements from several different surfaces with different material properties and by using a model of how the various parameters transform between different views. These allow functions to be derived that are independent of the unknown driving conditions. Some of these functions can be used as meaningful thermal invariants.

The first step is to rewrite equation 1 as $\mathbf{a}^T\mathbf{x} = 0$, where $\mathbf{x}$ includes the driving conditions and $\mathbf{a}$ includes everything else. Several means of achieving this have been proposed resulting in different invariants. The following was described in [Michel *et al.*, 1997]:

$$\begin{aligned} a_1 &= C_T & x_1 &= -\frac{dT_s}{dt} \\ a_2 &= k & x_2 &= \frac{dT_s}{dx} \\ a_3 &= -(T_s - T_{amb}) & x_3 &= h \\ a_4 &= -\sigma(T_s^4 - T_{amb}^4) & x_4 &= \epsilon \\ a_5 &= \cos\theta_I & x_5 &= W_I\alpha_s. \end{aligned} \tag{7}$$

The second step is to model the transformation of $\mathbf{a}$ between views. For five linearly independent vectors $\mathbf{a}_1, \ldots, \mathbf{a}_5$ formed from one view (image) and five corresponding vectors $\mathbf{a}'_1, \ldots, \mathbf{a}'_5$ from a second view, there is a unique matrix M, such that $\mathbf{a}'_j = M\mathbf{a}_j$, $j = 1, \ldots, 5$. By constraining the form of M based on prior knowledge the degrees of freedom of M can be reduced. Specifically, since C_T and k do not change between views, the first two rows of M are completely determined. Therefore, we have 25 constraints on M from the equations $\mathbf{a}'_j = M\mathbf{a}_j$ and only 15 unknowns. This implies that up to 10 functions involving only terms in the vectors $\mathbf{a}_j$ can be derived [Mundy and Zisserman, 1992] using elimination methods [Kapur *et al.*, 1995]. Since these functions are independent of imaging conditions, they should be invariant across different views. Some involve fewer than five points. Many are trivial, but others are potentially useful.

2.2 Invariants Tested

We have begun testing four invariants proposed in the literature. The first one, which is quite simple, was proposed in [Gauder *et al.*, 1993]. The other three are based on different formulations of $\mathbf{a}$ and $\mathbf{x}$ derived from the physical model of equation 1. These invariants, $I2, \ldots, I4$, are specialized to data taken at night, where $W_I = 0$, so that $\mathbf{a}$ has only four components. Each of these is a ratio of determinants.

I1: This is just a ratio of temperature differences [Gauder *et al.*, 1993]. Given three points (or regions), denoted by m, n, and p, with surface temperatures T_m, T_n, and T_p, the invariant is

$$I1(m,n,p) = \frac{T_m - T_n}{T_n - T_p}. \tag{8}$$

This invariant does not require prior knowledge of material properties.

I2: This invariant [Michel *et al.*, 1996] requires four points (regions), m, n, p and q, measured surface temperatures T_i, $i \in \{m,n,p,q\}$, and hypothesized material properties, k_i and $C_{T,i}$. The materials must be distinct. To simplify the notation, let $\mathbf{b}_i = (T_i, k_i, C_{T,i})'$ (where $'$ denotes "transpose" here). Then,

$$I2(m,n,p,q) = \frac{|\mathbf{b}_m, \mathbf{b}_n, \mathbf{b}_p|}{|\mathbf{b}_n, \mathbf{b}_p, \mathbf{b}_q|} \tag{9}$$

This does not require the ambient temperature.

I3: This invariant [Michel *et al.*, 1997] requires three points (regions), m, n, and p, measured surface temperatures, hypothesized material properties, and the ambient temperature, T_{amb}. Let $\mathbf{b}_i = (C_{T,i}, k_i, \sigma(T_{amb}^4 - T_i^4))'$, and let $\mathbf{c}_i = (C_{T,i}, T_{amb} - T_i, \sigma(T_{amb}^4 - T_i^4))'$. Then,

$$I3(m,n,p) = \frac{|\mathbf{b}_m, \mathbf{b}_n, \mathbf{b}_p|}{|\mathbf{c}_m, \mathbf{c}_n, \mathbf{c}_p|} \tag{10}$$

Observe that σ can be factored out of these calculations, which increases numerical stability.

I4: This invariant was suggested in personal communication by Greg Arnold. It requires four points (regions), m, n, p, and q, measured surface temperatures, the ambient temperature, and C_T but not k. Let $\mathbf{b}_i = (C_{T,i}, T_{amb} - T_i, \sigma(T_{amb}^4 - T_i^4))'$. Then,

$$I4(m,n,p,q) = \frac{|\mathbf{b}_m, \mathbf{b}_n, \mathbf{b}_p|}{|\mathbf{b}_n, \mathbf{b}_p, \mathbf{b}_q|} \qquad (11)$$

This invariant can be extended to use up to 6 points (regions) by replacing $\mathbf{b}_n$ and $\mathbf{b}_p$ with measurements from other points (regions).

3 Calculating and Using Thermal Invariants of Surface Regions

We envision the following complete scenario for using a thermal invariant to identify surface materials. The first three steps are currently implemented. The fourth and fifth are only partially implemented, but the implementation is complete enough for an initial study of the stability and relative performance of the invariants.

1. In one thermal image, the user outlines several regions (one less than needed to form the invariant) and chooses for each region the material type corresponding to the majority of the region's pixels. We refer to these known regions, which need not be completely homogeneous, as the "basis regions".

2. The user outlines a "test region" which has an unknown material type.

3. The user chooses a thermal invariant, one of $I1$ through $I4$.

4. If $I1$ is chosen, the system will search a database of invariants to select those corresponding to the basis region materials plus one more material. This will form a set of candidate materials and invariant values for the test region. Independent of this, the system will calculate the actual value of $I1$ for the basis regions and the test region over a sequence of thermal images. (This can be done independently because the invariant calculations do not depend on material parameters.) The actual invariant values will then be compared to the candidate invariant values to choose the correct material type or to determine that none are appropriate. This requires that the invariants for different materials be distinguishable.

5. If $I2$, $I3$ or $I4$ are chosen, the system must try different hypothesized materials for the test region. It will calculate invariant values for each hypothesized material over a sequence of thermal images. It will then decide the correct material for the test region in one of two ways. The first, which assumes that incorrect material hypotheses will produce fluctuating invariant values, is to choose the material corresponding to the most stable invariant. The second, which is more in-line with the use of $I1$, is to compare the invariant values against precomputed values for the tested combination of materials. The first represents the ideal case since it does not require building a database of invariants, but it also places the greatest demand on the accuracy of the model and the calculations.

When an invariant is being used for change detection, our envisioned scenario is much simpler. It requires only that the basis and test regions be identified (including perhaps their material types), that the invariant be calculated over an initial sequence of images to establish its range of values, and that the invariant be monitored in subsequent images for persistent and substantial changes.

Our implementation thus far has focused on experimentally evaluating invariants $I1$ through $I4$ using a variety of basis and test regions. The main parts of our implementation are discussed in the remainder of this section. The next section summarizes our preliminary experimental results.

3.1 Temperature Calibration

The raw MWIR data does not give surface temperature measurements without a calibration

model. For LWIR data, calibration is done using a physical model of the sensor together with assumptions about emissivity values of common materials [Nandhakumar and Aggarwal, 1988]. For the MWIR data we study here, the calibration model is entirely empirical [MTL Systems Inc., 1996]. Its derivation is based on comparing thermal image data to temperatures measured on the actual surface using thermocouples. This calibration does not lead to a physical understanding of the sensor.

3.2 Site Model and Camera Models

A crucial aspect of evaluating invariants over a sequence of thermal images is locating the same region in multiple thermal images. When these regions are parts of building walls, building roofs, roads or parking areas, a site model is needed. Hence, as part of this project we have been building a detailed site model of Wright-Patterson Air Force Base. We started from elevation data for some buildings and an initial CAD model containing building footprints. This gave enough information to build initial camera and building models.

New and more complete building models and improved camera models were then constructed iteratively. This was driven by manual identification of corresponding points in multiple visible light intensity images. This was all done within GE's TargetJr software system. Once complete the site model was used to construct camera models for the thermal images, forming the basis for subsequent research. An example site model overlaid on top of a MWIR image (after conversion to temperature measurements) is shown in Figure 1.

To form a basis region or a test region, the user starts by outlining a polygonal region of interest in one thermal image. The image coordinate vertices of this polygon are backprojected onto the site model to form a chain of vertices in world coordinates. (If the backprojection of a vertex does not intersect the site model, the region is rejected.) For a subsequent thermal image, these regions are projected to image coordinates using the camera model of the new

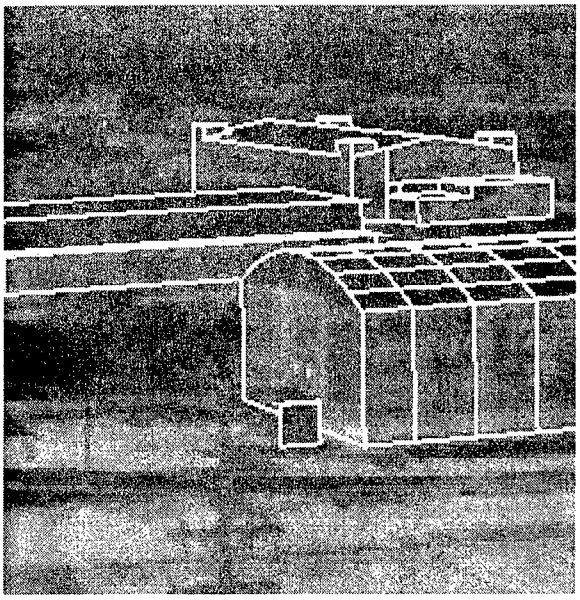


Figure 1: A thermal image with an overlaid site model

image, forming a new image region. Example basis regions and test regions are shown in Figures 2 and 3, respectively.

3.3 Temperature Calculation in Regions

Regions are used rather than individual points for two main reasons. First, since temperature measurements should be the same for different surface points having the same material properties, combining values from many pixels should produce more accurate temperature estimates and hence more accurate invariants. The trick is combining pixel values while tolerating values that may correspond to different materials (outliers). Second, individual points are difficult to locate in multiple views of a homogeneous region: these points are not distinctive and the camera models and site model are not precise enough to use backprojection and reprojection alone.

We estimate the temperature for an image region using a technique that tolerates up to 50% outliers but is more accurate than the median. The pixel values are gathered into a list and sorted into non-decreasing order. The resulting values may be denoted by $t[1], \ldots, t[n]$. Then, the smallest interval containing half the mea-

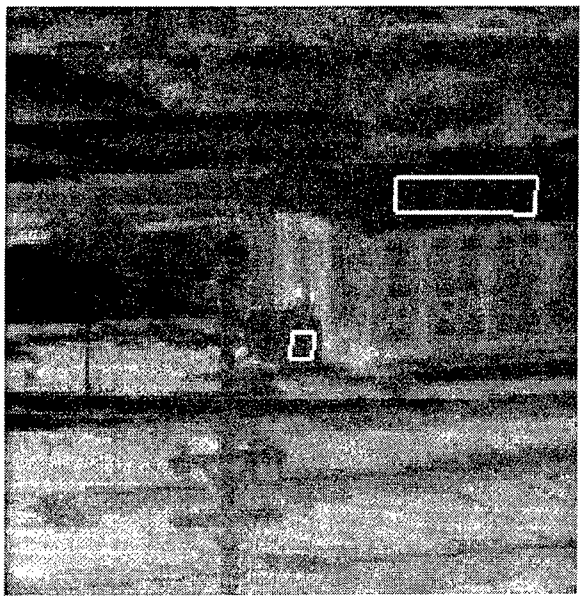


Figure 2: A thermal image with two overlaid, rectangular basis regions

Figure 3: A thermal image overlaid with a rectangular test region

surements is found. Letting $m = \lceil (n+1)/2 \rceil$, this interval starts at index i^*, where

$$i^* = \underset{i \in \{1,\ldots,n-m+1\}}{\operatorname{argmin}} (t[i+m-1] - t[i]).$$

An initial estimate of the temperature, which is already less biased than the median when the outlier distribution is skewed, is $t^* = (t[i+m-1] + t[i])/2$. An estimate of the scale which is consistent with Gaussian distributed temperature values is $s = 1.43 \cdot (t[i+m-1] - t[i])/2$. The final estimate of the temperature is found by gathering all points within the interval $t^* \pm 3s$ and computing the mean and standard deviation. This is much more accurate than the median but just as robust. The cost is sorting the data, which is trivial since typical regions contain 100 or so pixels.

3.4 Summary Statistics for An Image Sequence

Once the values are calculated for an invariant formed by a given set of basis regions and a test region across a sequence of images, summary statistics must be calculated. Currently, this is achieved using the technique described in Section 3.3. In effect, this treats the invariants as Gaussian, which they are unlikely to be. More sophisticated techniques, such as discussed in [Arnold *et al.*, 1996] may be needed when our software is extended to include decision techniques. For the initial experimentation described here, the current techniques are sufficient.

4 Preliminary Experimental Results

We have completed preliminary experimentation with the four invariants $I1$ through $I4$. These results explore the stability, accuracy and reliabilty of the invariants. We discuss each invariant in turn.

The test data were acquired at Wright-Patterson Air Force Base over a two week period in August of 1996. Thermal image data were taken at one hour intervals all through the day and night using a portable thermal imager mounted on top of a telescoping tower. The imager collected data from nine image segments.

Images from six twenty-four hour periods were acquired, with each segment being imaged every hour for each of the six periods. Ambient temperature readings were taken at 10 minute intervals during this period. In addition, visible light images were taken for use in building the site model. Our tests thus far have focused on images of the buildings shown in Figure 1. The center building has a fiberglass shingle roof, painted cinderblock walls, and galvanized steel in the arch between the cinderblock walls and the roof. Just below the building is a wooden trash screen. The parking lots in front of the center building are asphalt. A building in the background has face brick walls.

The most promising results thus far were obtained for invariant $I1$, a simple ratio of temperature differences. Figure 4 shows values of three different invariants over two nights, plotted as a function of Universal Time (subtract four hours to obtain Eastern Standard Time). The same basis regions of soft wood and cinderblock are used for all three invariants. The three test regions are fiberglass shingle, asphalt, and face brick. The plots show the invariant values to be relatively stable and somewhat separated from each other. These results are representative of other tests.

As further illustration of invariant $I1$, Figure 5 shows the values of the same invariant over a 24 hour period for three different days. The basis regions are fiberglass shingle and soft wood, and the test region is asphalt. The invariant value fluctuates radically during the day, but is steady at night.

The results for $I2$ are quite poor. The invariant values themselves are extremely stable, having standard deviations on the order of 10^{-3}. However, when the basis region set is fixed and the hypothesized material for the test region is fixed but the location and material type of the test region are shifted, the value for the invariant does not change significantly. This is consistent across all combinations of basis regions and test regions we have examined. Thus we have seen no discriminatory power in $I2$.

The results for invariants $I3$ and $I4$ are somewhat better. Representative results are shown

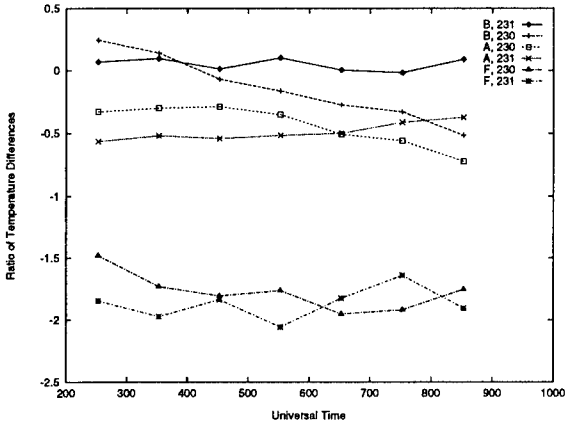


Figure 4: Plot of invariant values for $I1$ (or "ratio of temperature differences") against universal time for three different invariants over two nights. "F" indicates the fiberglass shingle test region, "A" indicates the asphalt test region, and "B" indicates the face brick test region. 230 and 231 are the Julian days for which the invariant values are plotted.

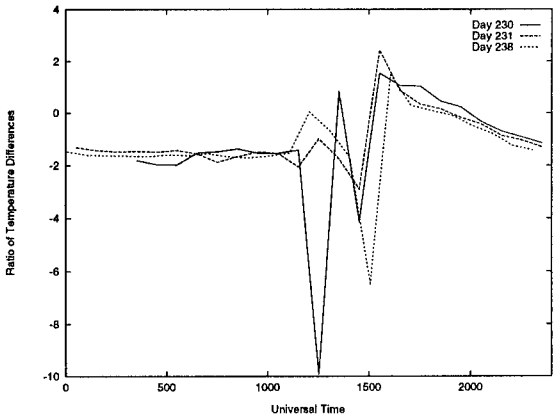


Figure 5: Plot of invariant values for $I1$ (or "ratio of temperature differences") against universal time for three different 24 hour periods.

in Tables 1 and 2. The results show a fixed set of basis regions, a series of test regions with known material types, and different hypothesized test region materials. Results are rank-ordered by the robust standard deviations of the invariant values. In some cases the correct region material yields the lowest standard deviation, while in others the correct material is no better and sometimes worse than the other materials. For invariant $I4$ concrete and cinderblock yield the lowest standard deviations consistently with the basis regions used here, but this effect does not carry over to other basis regions (data not shown).

The results for invariants $I2$, $I3$ and $I4$ are cause for concern. Since other work on thermal invariants has reported much more positive results [Arnold *et al.*, 1996, Michel *et al.*, 1997, Nandhakumar and Velten, 1994, Nandhakumar *et al.*, 1996], our guess, based on our initial results, is that there is a problem with the data, the material properties or the physical model.

- There are two potential problems with the data. One is the mapping from MWIR intensities to temperatures, which is based on a heuristic calibration equation rather than a physical model. The second is that the ambient temperature readings, which are only measured to tenths of degrees, appear to fluctuate.

- Material properties k and C_T are difficult to obtain. The method used in other studies is to match material descriptions with entries in tables of materials printed in texts such as [Incropera and DeWitt, 1981]. This is particularly difficult for the painted surfaces and building materials which predominate in our image data.

- There are two potential problems with the physical model. The first is that the model was derived for LWIR imagery whereas the current data is MWIR imagery. This means the validity of the physical model has not yet been tested. The second potential problem is that the model assumes the only incident heat is solar radiation, which is zero at night. However, if there is heat-

Tests		Standard
Region	Material	Deviation
Soft wood	Soft wood	4.03
	Fiberglass	5.59
	Concrete	15.1
	Rubber	29.3
	Cinder	32.2
	Steel	37.9
	Brick-common	41.1
	Tin	1550
Cinder	Brick-common	5.30
	Cinder	5.61
	Soft wood	6.36
	Fiberglass	8.32
	Concrete	27.1
	Rubber	28.0
	Steel	40.7
	Tin	661
Steel	Fiberglass	2.90
	Soft wood	5.88
	Concrete	5.96
	Rubber	22.3
	Steel	34.5
	Brick-common	36.9
	Cinder	37.6
	Tin	2160
Fiberglass	Concrete	6.38
	Steel	8.62
	Cinder	10.5
	Brick-common	13.6
	Rubber	13.8
	Fiberglass	19.2
	Soft wood	34.8
	Tin	316

Table 1: Sample test results for invariant $I3$ using face brick and asphalt as basis regions. The test region was varied and tested against a battery of materials. Results for each test region are ranked in increasing order of standard deviation, to three significant figures.

Tests		Standard
Region	Material	Deviation
Cinder	Concrete	0.0933
	Cinder	0.200
	Steel	0.779
	Brick-common	0.810
	Rubber	5.83
	Tin	6.07
Steel	Concrete	0.174
	Cinder	0.506
	Steel	0.896
	Tin	0.950
	Rubber	1.09
	Brick-common	1.42
	Soft wood	16.8
Soft wood	Concrete	0.101
	Cinder	0.390
	Rubber	0.569
	Steel	0.896
	Tin	0.950
	Brick-common	1.08
	Soft wood	15.8

Table 2: Sample test results for invariant $I4$ using face brick, asphalt and fiber glass as basis regions. The test region was varied and tested against a battery of materials. Results for each test region are ranked in increasing order of standard deviation, to three significant figures.

ing (or cooling) from inside the buildings, this assumption is violated.

If further experimentation with the current data shows results similar to those reported here, these concerns should be investigated.

5 Summary

We have reported our current and ongoing work in using thermal invariants to identify materials and to monitor changes in materials in sequences of thermal imagery. We have built a detailed site model, developed camera projection models, and implemented a system to calculate invariants from a user selected set of regions over the image sequence. Our initial experimental results are mixed, with some invariants working relatively well and others poorly. Our ongoing work will mostly be experimental, with further analysis to identify the cause of problems if they persist. We hope these studies will enable us to develop a robust system for classifying materials based on thermal invariants, both to identify material types and to monitor sites of interest for changes in materials over time.

Acknowledgements

The authors would like to thank Greg Arnold for his help with various aspects of this work.

References

[Arnold *et al.*, 1996] D.G. Arnold, J. Michel, N. Nandhakumar, G. Tsihrintzis, and V. Velten. Robust thermophysics-based interpretation of radiometrically uncalibrated IR images for ATR and site change detection. In *Proceedings of the DARPA Image Understanding Workshop*, 1996.

[Gauder *et al.*, 1993] M. Gauder, V. Velten, L. Westerkamp, J. Mundy, and D. Forsyth. Thermal invariants for infrared target recognition. In *Proceedings of the ATR Systems and Technology Conference*, 1993.

[Incropera and DeWitt, 1981] F.P. Incropera and D.P. DeWitt. *Fundamentals of Heat Transfer*. John Wiley and Sons, 1981.

[Kapur *et al.*, 1995] D. Kapur, Laksman Y. N. and T. Saxena. Computing invariants using elimination methods. In *Proceedings of the International Symposium on Computer Vision*, 1995.

[Michel *et al.*, 1996] J.D. Michel, N. Nandhakumar, T. Saxena, and D. Kapur. Using elimination methods to compute thermophysical algebraic invariants from infrared imagery. In *Proceedings of the American Association for Artificial Intelligence*, 1996.

[Michel *et al.*, 1997] J. Michel, N. Nandhakumar, and V. Velten. Thermophysical algebraic invariants for infrared imagery for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):41–51, 1997.

[MTL Systems Inc., 1996] MTL Systems Inc. Ground truth data collection report. Technical report, October 1996.

[Mundy and Zisserman, 1992] J.L. Mundy and A. Zisserman, editors. *Geometric Invariance in Computer Vision.* MIT Press, 1992.

[Nandhakumar and Aggarwal, 1988] N. Nandhakumar and J.K. Aggarwal. Integrated analysis of thermal and visual images for scene interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):469–481, 1988.

[Nandhakumar and Velten, 1994] N. Nandhakumar and V. Velten. Thermophysical affine invariants for ir imagery for object recognition. In *Proceedings of the DARPA Image Understanding Workshop*, pages 1403–1412, 1994.

[Nandhakumar *et al.*, 1996] N. Nandhakumar, J. Michel, D.G. Arnold, G. Tsihrintzis, and V. Velten. Robust thermophysics-based interpretation of radiometrically uncalibrated IR images for ATR and site change detection. *IEEE Transactions on Image Processing*, 1996.

Texture Segmentation of SAR Images

By-Her Wang and Thomas O. Binford *

Robotics Laboratory, Computer Science Department
Stanford University, Stanford, CA 94305

Abstract

The image surface of synthetic aperture radar imagery (SAR) is dominated locally by peaks and clusters of peaks, especially in terrain and vegetation areas. More globally, there are extended regions distinguishable by texture, trees, fields, shadows, roads, etc. We describe an algorithm which segments SAR images into a set of regions of pre-specified classes, based on two procedures: first, the classification of peaks into $N = 3$ pre-specified classes, and second, a segmentation of the Delaunay triangulation of peaks into connected regions. A peak detection operator is used to estimate peaks in SAR images; thresholds are determined by using the histogram of the peak amplitude of each class. Peak amplitude was found to be the most useful discriminant by far in the multi-variate distribution in peak amplitude, peak width, and peak density. A Delaunay triangulation was established on the peaks of each class. Links in the triangulation were removed if they were unlikely for a population of that class. The boundary of a texture region is the boundary of a connected component of the modified Delaunay triangulation of the appropriate class of peaks. Linking by boundary traversal was developed to extract closed boundaries of each class. Experimental and simulation results are presented in SAR and synthesized images, respectively. Boundaries of regions can be determined to an accuracy of about 2 pixels.

1 Introduction

One apparent characteristic of Figure 3 is the density of peaks nearly everywhere in the image. From the physics of SAR imaging, trihedrals are important in analyzing scattering from components of targets and in clutter analysis. Around peaks, the SAR impulse can be modeled by two-dimensional Gaussian functions. It is known that other components have different scattering behaviors, e.g. dihedrals. An image may be interesting at the level of single peaks, e.g. a corner reflector may be interesting on its own. However, typically, physically similar areas are interesting as extended structures or regions, e.g. trees, grass, fields, buildings, and targets. Extended regions appear as textures of peaks.

Thus, a second level of image model is piecewise regions of uniform textures of peaks. Textures could be complex, directional, and hierarchical; in fact they are complex, as in a plowed field. But a class of textures appears useful that are isotropic, i.e. non-directional, distinguished by the values of peak amplitude, peak width, and peak density. Further, it appears empirically as though the local textures are distinguished by the probability density function (pdf) of peak amplitude.

There is a growing interest in the development of algorithms which can extract boundaries automatically for a broad range of applications, such as conventional optical, radar and medical imagery [5]-[14]. In some domains, boundary detectors identify boundaries between surfaces with uniform reflectivity as oriented edgel discontinuities of order zero or one in the image intensity surface (*e.g.* [1]). Some special SAR images containing smooth objects (*e.g.* sea ice) can be segmented by an edgel-based operator[2]. Unfortunately, this kind of edgel-based algorithm can not be applied to general SAR images very generally because various features, either natural or man-made, are dominated by peaks.

A peak-based texture segmentation operator is thus important for SAR images. This paper develops a new algorithm to extract closed boundaries of various features in SAR images. N classes of peaks, in our case tree, ground, and shadow, are extracted by a peak detector [3] and classified by thresholds. For an image including one single class of peaks, peak densities can be determined from distances of neigh-

*This research was supported by a contract from the Air Force, F33615-93-1-1281 through WPAFB from ARPA ASTO "Multi-Sensor ATR: Quasi-Invariants and High Accuracy Measurements in Bayesian Inference" and "Context and Quasi-Invariants in ATR with SAR Imagery".

bors, thus from lengths of links in the Delaunay triangulation; from this it determines discontinuities in peak density. Therefore, boundaries can be determined by traversal of peaks which are located at these discontinuities.

In Section 2, peak detection and classification are described briefly. In Section 3, the algorithm for texture segmentation is presented. In Section 4, the performance of this operator is examined in synthesized and real SAR images.

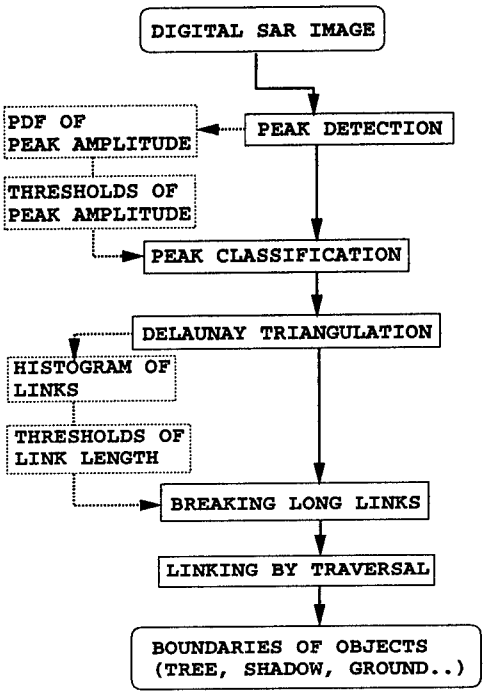


Figure 1. Block diagram of the texture segmentation algorithm.

2 Peak Detection and Classification of Peaks

A peak detector developed in previous work[3] is used to extract peak position, width and amplitude in SAR images. Based on pdfs of peak amplitude, peak width, and peak spacing for pre-specified classes of peaks, thresholds are determined interactively to distinguish groups of peaks, in our case, shadow, trees, and grass.

2.1 Peak Detection

Extended elements in SAR images consist of many narrow peaks generated from imaged point scatterers, with the SAR impulse modeled by two-dimensional Gaussian functions[3]. Experimentally, the assumption of separable Gaussian peaks (see Eq.(1)) for narrow peaks is correct.

$$I(x,y) = He^{-\frac{x^2}{2\sigma_x{}^2} - \frac{y^2}{2\sigma_y{}^2}} \tag{1}$$

After applying a normalized Gaussian filter, the filtered peak can be written as

$$f(x,y) = \frac{H\sigma_x\sigma_y}{\sqrt{(\sigma_x{}^2+\sigma_t{}^2)(\sigma_y{}^2+\sigma_l{}^2)}} e^{-\frac{x^2}{2(\sigma_x{}^2+\sigma_t{}^2)} - \frac{y^2}{2(\sigma_y{}^2+\sigma_l{}^2)}} \tag{2}$$

where

σ_t = *mask width in transverse (or x) direction*

σ_l = *mask width in longitudinal (or y) direction*

For a separable two-dimensional Gaussian function, $f(x,y)$, the level lines of $f_x = 0$ and $f_y = 0$ are parallel to the x-axis and y-axis, respectively. Therefore, the peak position can be determined by a unique point that satisfies both conditions of $f_x = 0$ and $f_y = 0$ inside the 3×3 grid of pixels.

Peak width, σ_x or σ_y, can be obtained from quadratic equations (see Eqs.(3a) and (3b)) which are derived by Eq.(2) and its second derivative.

$$\frac{f_{xx}(x,y)}{f(x,y)} = \frac{x^2}{(\sigma_x{}^2+\sigma_t{}^2)^2} - \frac{1}{(\sigma_x{}^2+\sigma_t{}^2)}$$

or

$$[\frac{f_{xx}}{f}](\sigma_x{}^2+\sigma_t{}^2)^2 + (\sigma_x{}^2+\sigma_t{}^2) - x^2 = 0 \tag{3a}$$

$$[\frac{f_{yy}}{f}](\sigma_y{}^2+\sigma_l{}^2)^2 + (\sigma_y{}^2+\sigma_l{}^2) - y^2 = 0 \tag{3b}$$

Peak Amplitude, H, can be solved by substituting peak width back to Eq.(2).

2.2 Peak Classification

The peak-based classification algorithm is described in this subsection. To illustrate, detected peaks in a SAR sub-image are shown in Figure 3(c), and peaks of the non-shadow region are presented as black dots in Figure 3(e).

- Histogram Analysis and Thresholding
  A peak detector is applied to selected training SAR images with N classes, respectively. From the detected peaks, the histogram and pdf of peak amplitude for each class is computed. For N pre-specified classes, $(N-1)$ thresholds can be set in order to segment N regions. For example, three classes are considered in this case, *i.e.* tree, ground, and shadow. Based on Bayes' decision rule, two thresholds are chosen from the pdf of peak amplitude to separate peaks in tree regions and peaks in non-shadow regions, respectively.

- Peak Classification
  Given a digital SAR image, we detect peaks at different amplitudes by using the peak detector described in Section 2.1. The first threshold is used to extract peaks of tree regions and the second threshold is used to extract peaks of non-shadow regions.

3 Texture Segmentation Algorithm

After peak classification, $(N-1)$ images can be obtained for an original image with N classes. The peak density can be determined by lengths of the links in the Delaunay triangulation. A histogram of link lengths can be generated to obtain a threshold. The hypothesis is that after breaking longer links, the boundary peaks (compared to edge pixels) can be extracted from discontinuities in the peak density. A linker by traversal connects boundary peaks to determine the closed boundary. This whole algorithm is shown in Figure 1 and described in the following three stages:

- Stage 1: Delaunay Triangulation
  A triangulation of P is defined as a maximal planar subdivision to interpolate a terrain given P sample of points[4]. Given a set P of points in the plane, any locally and globally equiangular triangulation of P is the Delaunay triangulation of P. Therefore, the optimal approximation of a terrain can be achieved by the Delaunay triangulation. One example of the Delaunay triangulation is shown in Figure 3(d) for the terrain of non-shadow.

- Stage 2: Breaking Long Links
  The lengths of sides of triangles in the Delaunay triangulation is inversely proportional to the peak density locally as shown in Figure 3(d). The histogram of lengths of sides of all triangles from the Delaunay triangulation is computed to determine the threshold interactively. Links that are longer than the threshold are broken and recorded. It is intuitively clear that the boundary points can be obtained by collecting the points that have broken links. Those points are discontinuities in peak density.

- Stage 3: A Linker by Traversal
  To connect those boundary peaks extracted by the last stage, a linker by traversal is used. In Figure 2, locally a vector, centered at one boundary peak (black dot) and pointed at the previous neighboring boundary peak (gray dot), is rotated by the minimum angle to the next neighboring boundary peak (gray dot). Boundary peaks are linked by this simple traversal algorithm. Finally, with knowledge of the average peak distance, the boundary can be determined by extending the linked curve by half of the average peak distance. An example is shown in Figure 3(e) for regions of shadow.

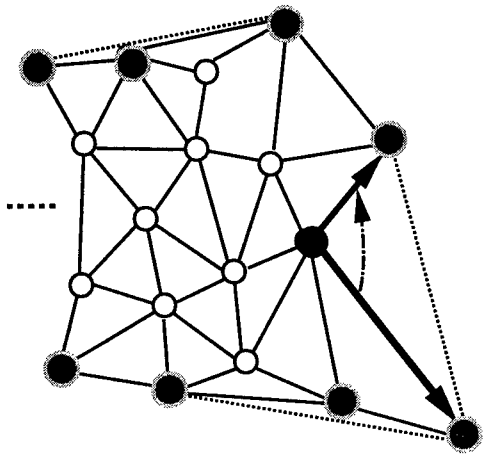


Figure 2. An example of linking by traversal, where the dashed lines represent the broken links and solid lines are shorter than the threshold.

4 Simulation and Experimental Results

The performance of this new texture segmentation operator can be evaluated by simulation results. Each test image is synthesized with two concentric square regions, *i.e.* shadow and non-shadow regions. The boundary of shadow is estimated and compared with the pre-specified boundary. The average estimation error is less than 0.2 pixel at peak density of 0.11 per pixel; the standard deviation of estimation is 1.5 pixels. At peak density of 0.14 per pixel, the average estimation error is less than 0.4 pixel and the standard deviation is 1.0 pixel. For real SAR images that we are deal with, the average peak density is about 0.125 per pixel resulting from the peak detector. Therefore, simulation results show

that this algorithm determines boundaries to an accuracy of about 2 pixels in SAR images. The experimental result is demonstrated in Figure 4, which shows the boundaries of tree and shadow regions in a sub-image of Figure 3(a).

5 Conclusion

A peak-based texture segmentation operator has been developed to extract boundaries of regions with uniform textures in SAR images. Peak classification was accomplished with the pdf of peak amplitude for every pre-specified class. With the Delaunay triangulation, discontinuities in peak density can be determined by links of triangles. Thus, the boundary peaks can be extracted at discontinuities and linked by traversal. Finally, the boundaries can be refined by extending these curves by half of average peak distance.

The performance of this new operator is demonstrated by simulation and experimental results. Simulation results show that this algorithm determines boundaries to an accuracy of about 2 pixels.

References

[1] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp.679-698, 1986.

[2] J. Helterbrand, "One-Pixel-Wide Closed Boundary Identification," *IEEE Trans. Image Processing*, vol. 5, no. 5, pp.780-783, 1996.

[3] B. Wang and T.O. Binford, "Generic, Model-based Estimation and Detection of Peaks in Image Surfaces," *Proceedings of Image Understanding Workshop*, Vol. 2, pp.913-922, 1996.

[4] M. Berg *et. al.*, "Computational Geometry by Example," *Department of Computer Science, Utrecht University, the Netherlands*, Chapter 9, pp.159-171, 1996.

[5] H. Voorhees and Tomaso Pogglo, "Computing texture boundaries from images," *Nature*, vol.333, no.6171, pp. 364-367, 1988.

[6] J. R. Bergen, "Theories of Visual Texture Perception," *Spatial Vision", D. Regan, ED., CRC PRESS*, 1991.

[7] M. Unser, "Texture Classification and Segmentation Using Wavelet Frames," *IEEE Trans. on Image Processing*, vol. 4, no.11, pp.1549-1560, 1995.

[8] A. Talukder *et. al.* "Model selection and texture segmentation using partially ordered Markov models," *1995 International Conference on Acoustics, Speech, and Signal Processing*, vol.4 , pp. 2527-30, 1995.

[9] H. Chao *et. al.* "Texture segmentation using joint time frequency representation and unsupervised classifier," *1995 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp.304-309, 1995.

[10] X. Xie *et. al.* "Texture segmentation using visual nonlinearity," *SPIE*, vol. 2597, pp.142-149, 1995.

[11] D. Dunn and W. E. Higgins, "Optimal Gabor filters for texture segmentation," *IEEE Trans. on Image Processing*, vol. 4, no. 7, pp.947-964, 1995.

[12] D. J. Telfer and K. O. Pritchard, "Histogram correlation of the output from a small mask operator: a basis for adaptive texture segmentation," *Fifth International Conference on Image Processing and its Applications*, no. 410, pp. 841-846, 1995.

[13] K. Sutherland and J.W. Ironside, "AUTOMATIC TEXTURE SEGMENTATION USING MORPHORLOGICAL FILTERING ON IMAGES OF THE HUMAN CEREBELLUM," *Proceedings of the 5th International Conference on Image Processing and its Applications*, no.410, pp. 777-780, 1995.

[14] S. Krishnamachari and R. Chellappa, "Multiresolution Gauss-Markov random field models for texture segmentation," *IEEE Trans. on Image Processing*, vol.6, no.2, pp. 251-67, 1995.

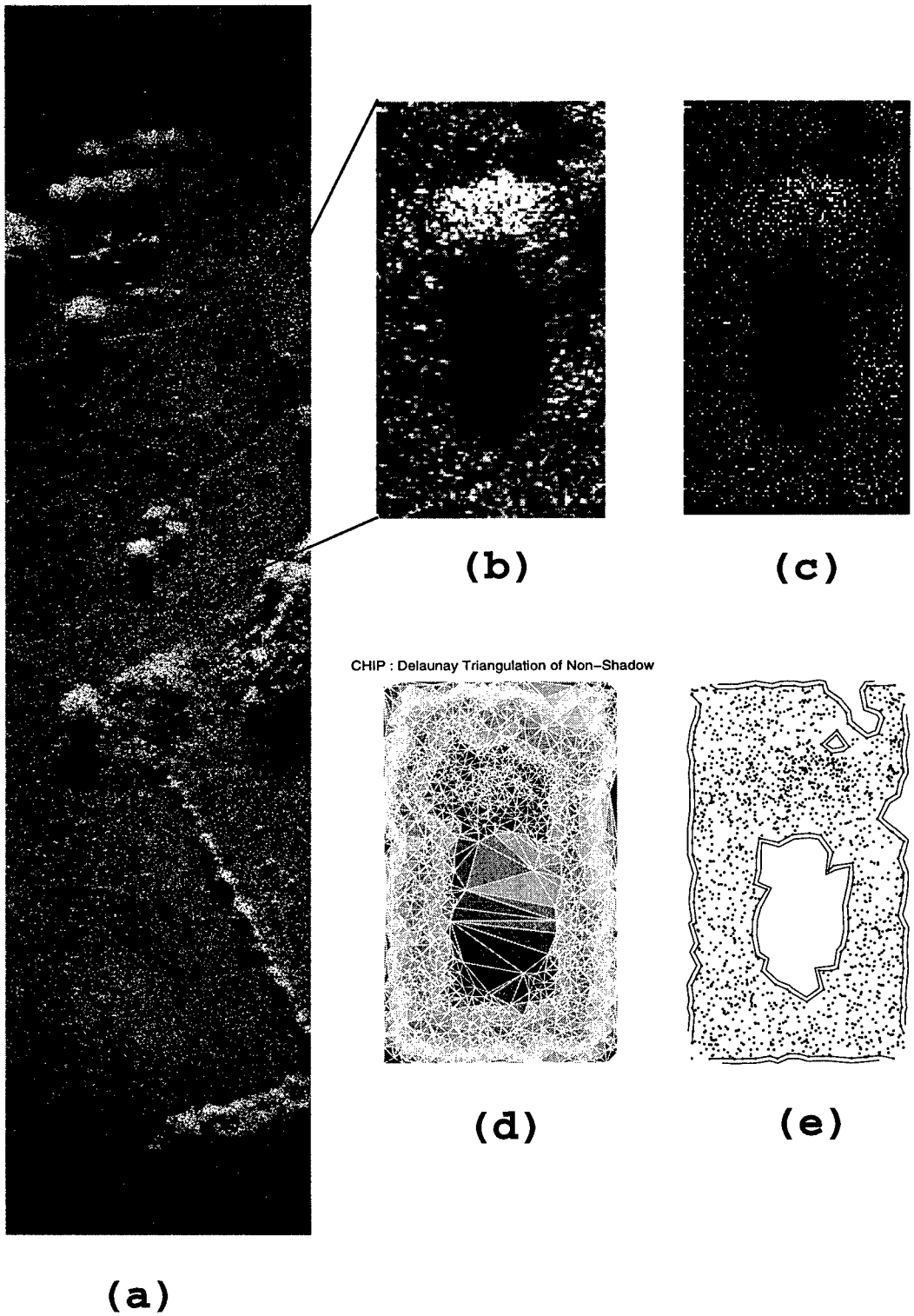


Figure 3: Structures in SAR image: (a) Part of a SAR image with sub-image indicated; (b) a sub-image magnified with features (tree, ground, and shadow); (c) detected peaks in sub-image; (d) Delaunay triangulation of peaks in non-shadow population; (e) boundary of shadow.

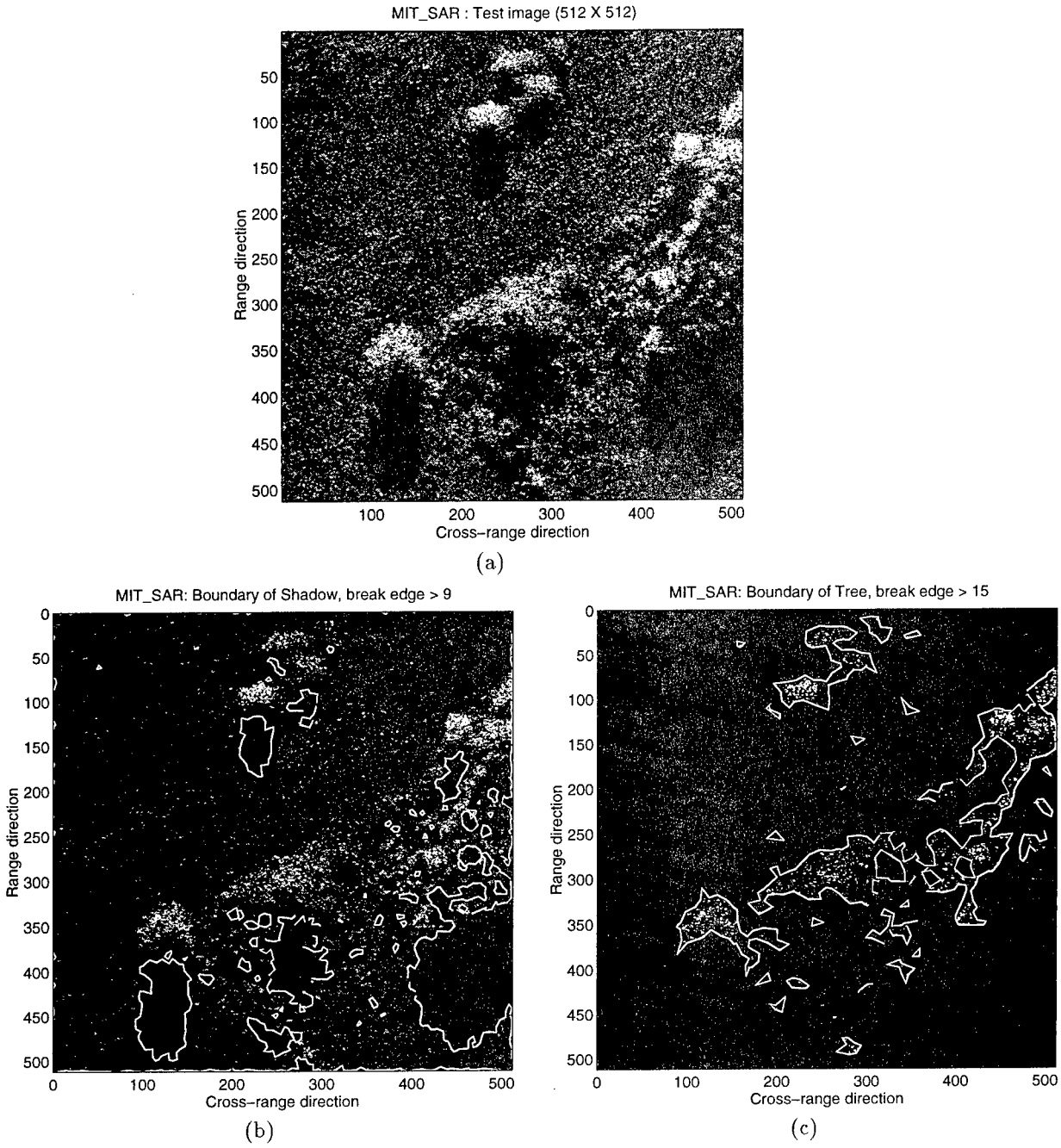


Figure 4: An example of texture segmentation in SAR image (a) Part of a SAR image (512 X 512); (b) boundaries of shadow; (c) boundaries of tree.

Sketching Natural Terrain from Uncalibrated Imagery

Q.-T. Luong*

Artificial Intelligence Center, SRI International
333 Ravenswood Ave., Menlo Park, CA 94025 USA

Abstract

We propose a methodology to sketch the 3D geometry of an outdoor scene consisting of natural terrain. The method requires only a pair of uncalibrated images, but it produces a sketch where the order with respect to the dimensions of height above the ground plane and depth are correct. A dense representation is generated as a set of profile lines which overlays the original images.

1 Introduction and related work

Terrain reconstruction is an important task in computer vision. There has been an extensive amount of work done in this area with calibrated cameras. The most successful approaches use stereo rigs or multiple views. However, there are many situations where the calibration data (camera parameters and relative position and orientation of the cameras) is not available. These situations are becoming more important as computer vision applications are no longer limited to robotics. In this context, it is also desirable to have an approach which can be used with a minimal number of views.

Although the investigation of the capacities of uncalibrated systems has become a popular research topic over the past few years (see references in [Luong and Faugeras, 1996]) there are very few systems which are actually able to perform a usable three-dimensional reconstruction. A first class of approaches is to perform a projective reconstruction from two views [Faugeras, 1992, Hartley *et al.*, 1992, Shashua and Navab, 1994] . However, the amount of deformation can be very large, which limits the usefulness of the reconstruction. A second class of approaches is to recover metric representations by performing self-calibration [Luong and Faugeras, 1997, Hartley, 1994] (which requires a large number of views to obtain stable results) or by using scene knowledge [Boufama *et al.*, 1993] . The last approach has been used with success to reconstruct buildings from multiple images [Faugeras *et al.*, 1995]. It requires line segments and geometric constraints which are not available in images of natural terrain.

It is generally believed that the area-based approaches to stereo are the most adequate for natural terrain, since well-defined geometric features are generally lacking. These approaches produce a depth map from which further processing is necessary in order to extract higher level information about the terrain. By contrast, we propose to represent the terrain by a set of profile lines, which is the trace of the terrain surface on a plane in 3D at the given depth.

*This work was sponsored by the Defense Advanced Research Projects Agency under contract DACA76-92-C-008 monitored by the U.S. Army Topographic Engineering Center, Alexandria, VA. The author also acknowledges Martin Fischler for discussions and the Robotvis group at I.N.R.I.A. for providing software tools. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Advanced Research Projects Agency, the United States Government, or SRI International.

This representation has a direct meaningful interpretation.

The classical approach to stereo consists in determining the disparity for each point along an epipolar line. The epipolar line is determined only by the geometry of the cameras, and within this line, each point corresponds to a different depth. By contrast, given a fixed depth, we propose to find all the points which lie at this depth. This is based on the idea that for the points which lie at a fixed depth, there is an analytical relation between their projections in multiple images, as exploited for stereo in the calibrated case [Robert *et al.*, 1992, Robert and Deriche, 1996], and for relative positioning of pairs of points in the uncalibrated case [Robert and Faugeras, 1993]. By sweeping 3D space with planes at a set of different depths, a representation of the terrain is obtained. The idea of the sweeping plane method was presented in [Collins, 1996], where it was argued that such a technique makes a full and efficient use of multiple images. Our work extends these ideas in two important directions. First, we show that in order to generate a qualitatively useful elevation map, full calibration of the cameras is not necessary. Instead, the only requirement, in addition to the epipolar geometry, is the identification of correspondences on the horizon, a technique well adapted to the type of scenes we consider. This makes it possible to apply our technique with as few as two uncalibrated views. Second, we propose a method based on curve evolution to generate the profile lines. This makes it possible to enforce continuity, smoothness, and uniqueness constraints in space (unlike in [Collins, 1996], which was based on geometric primitives). The use of PDE methods and level set implementations to identify the trace of a surface on a given plane has been explored by [Deriche *et al.*, 1996]. This work was theoretically extended to 3D and multiple cameras by [Faugeras and Keriven, 1996], however these authors demonstrate only a 2D implementation in the calibrated case. Our work differs from the above references by explicitly addressing the situation of a natural scene viewed from ground level. Thanks to the exploitation of fairly general domain-specific constraints, we obtain a semantically meaningful representation, a simpler propagation scheme, and we address problems caused by the fact that the surfaces are subject to occlusions.

2 Affine calibration

Affine calibration of a pair of views consists in determining enough geometric parameters for this pair of views so that the ambiguity in reconstruction will be at most an affine transformation of space. This is intermediate between the usual Euclidean calibration (which cannot be performed in our application), and the projective calibration which is too weak. In this section, we first describe the theory of affine calibration, and then detail our practical approach.

2.1 Fundamental matrix and infinity homography

The projective model We use the pinhole model. The main property of this camera model is that *the relationship between the world coordinates and the pixel coordinates is linear projective*: non-linear optical distortions are neglected. The consequence is that the relationship between 2-D pixel coordinates and any 3-D world coordinates can be described by a 3×4 matrix $\tilde{\mathbf{P}}$, called projection matrix, which maps points from $\mathcal{P}^3$ to $\mathcal{P}^2$:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \tilde{\mathbf{P}} \begin{bmatrix} \mathcal{X}_1 \\ \mathcal{X}_2 \\ \mathcal{X}_3 \\ \mathcal{X}_4 \end{bmatrix} \quad (1)$$

where the retinal projective coordinates x_1, x_2, x_3 are related to usual pixel coordinates by $(u, v) = (x_1/x_3, x_2/x_3)$ and the projective world coordinates $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$, $\mathcal{X}_4$ are related to usual affine world coordinates by $(X, Y, Z) = (\mathcal{X}_1/\mathcal{X}_4, \mathcal{X}_2/\mathcal{X}_4, \mathcal{X}_3/\mathcal{X}_4)$. The points for which $\mathcal{X}_4 = 0$ cannot be related to affine space, and are called *points at infinity*. They define a plane which is called *plane at infinity*.

The projection matrix $\tilde{\mathbf{P}}$ can be decomposed uniquely as:

$$\tilde{\mathbf{P}} = \mathbf{A}[\mathbf{R}, \mathbf{T}]$$

The matrix $\mathbf{A}$ has five entries which are called

intrinsic parameters (which describe characteristics of the camera and digitalization system such as focal length, aspect ratio). $\mathbf{R}$ and $\mathbf{T}$ called *extrinsic parameters* describes the change of world coordinate system (the pose of the camera). A camera is said to be calibrated when its intrinsic parameters are known, which makes it possible to derive metric measurements. Camera calibration requires the observation of a number of control features with known 3D coordinates. For this reason, it is not always possible to have access to this information, and this is why we propose a method which can deal with uncalibrated cameras.

Projective calibration: the fundamental matrix When considering two projective views, the main geometric property is known in computer vision as the epipolar constraint. It can be shown only from the hypothesis (1) that the relationship between the projective retinal coordinates of a point $\mathbf{m}$ and the projective coordinates of the corresponding epipolar line $\mathbf{l}'_m$ is linear. The *fundamental matrix* [Luong and Faugeras, 1996] describes this correspondence:

$$\begin{bmatrix} l'_1 \\ l'_2 \\ l'_3 \end{bmatrix} = \mathbf{l}'_m = \mathbf{F}\mathbf{m} = \mathbf{F} \begin{bmatrix} x_1 \\ x_2 \\ x_2 \end{bmatrix}$$

The epipolar constraint has then a very simple expression: since the point $\mathbf{m}'$ corresponding to $\mathbf{m}$ belongs to the line $\mathbf{l}'_m$ by definition, it follows that

$$l'_1 x'_1 + l'_2 x'_2 + l'_3 x'_3 = \mathbf{m}'^T \mathbf{F}\mathbf{m} = 0 \qquad (2)$$

The epipolar transformation is characterized by the 2×2 projective coordinates of the epipoles $\mathbf{e}$ and $\mathbf{e}'$ (which are defined respectively by $\mathbf{F}\mathbf{e} = 0$ and $\mathbf{F}^T\mathbf{e}' = 0$), and by the 3 coefficients of the homography between the two pencils of epipolar lines. These seven independent parameters represent the only generic information relating two uncalibrated views. Unless further hypotheses are made, there is no way to extract other geometric parameters from correspondences.

The Fundamental matrix of a pair of images can be computed using only point correspondences thanks to Eq. (2). Robust methods were studied in [Zhang *et al.*, 1995], including automatic generation of correspondences.

Affine calibration: the infinity homography matrix The fundamental matrix describes the *projective* geometry of the system of two cameras. Knowing only the fundamental matrix makes it possible to perform a 3D reconstruction up to a general projective transformation of space. Such a representation is not very useful for sketching purposes, because the degree of deformation can be very large. With general projective transformations, there is in particular no guarantee that inversions with respect to the depth do not occur.

A way to avoid these problems is to perform an affine calibration of the pair of cameras [Quan, 1993, Luong and Viéville, 1996]. This means that in addition to the fundamental matrix, we need to identify the homography $\mathbf{H}_\infty$ of the plane at infinity, defined as follows: the projective coordinates of two points $\mathbf{m}$ and $\mathbf{m}'$, projections in the first and second image of a point at infinity, are related by:

$$\mathbf{m}' \simeq \mathbf{H}_\infty \mathbf{m} \qquad (3)$$

Since any homography matrix satisfies $\mathbf{H}\mathbf{e} \simeq \mathbf{e}'$, in theory we need only the correspondence of three points in order to be able to compute the matrix $\mathbf{H}_\infty$.

2.2 Affine calibration in practice

Determining points at infinity In structured scenes, a method which works is to consider vanishing points. A vanishing point is the projection of a virtual point at infinity, defined by the convergence of parallel lines in space. In a natural scene of the type we are interested in, a more appropriate method is to identify corresponding points at the horizon.

This can be done using a set of simple, but efficient heuristics, as shown by Fischler in [Fischler, 1996], where a method to extract the skyline was described. Strictly speaking, there is no guarantee that the skyline represents the horizon, however, in practice, given the image res-

olution, points do not need to lie very far from the camera to be considered at infinity.

Robust computation of the infinity homography Once the Fundamental matrix is determined, there are only three degrees of freedom for the infinity homography. These three degrees of freedom can be represented by the vector $\mathbf{r}$ [Luong and Viéville, 1996] such that:

$$\mathbf{H}_\infty = [\mathbf{e}']_\times \mathbf{F} + \mathbf{e}'\mathbf{r}^T \quad (4)$$

where the symbol $[.]_\times$ designates the skew-symmetric matrix associated to the cross-product. Writing $\mathbf{H}_\infty$ under this form ensures that this matrix is actually consistent with the fundamental matrix, in the sense that any pair of the form $(\mathbf{m}, \mathbf{H}_\infty\mathbf{m})$ will satisfy the epipolar constraint in Eq. (2).

In order to compute robustly $\mathbf{H}_\infty$, it is necessary to use more than three points. By substitution of Eq. (3) into Eq. (4), each of the correspondences yields one equation (the two other equations are proportional) which is linear in $\mathbf{r}$. These equations are solved by a linear least-squares method, to obtain a starting point for the final non-linear minimization, in which the vector $\mathbf{r}$ is determined by minimizing the least-squares sum of the error terms, for each correspondence $(\mathbf{m}_i, \mathbf{m}'_i)$ of the horizon line:

$$d(\mathbf{m}'_i, \mathbf{H}_\infty\mathbf{m}_i) + d(\mathbf{m}_i, \mathbf{H}_\infty^{-1}\mathbf{m}'_i)$$

where $\mathbf{H}_\infty$ is given by Eq. (4) and $d(.,.)$ is the Euclidean distance between 2D points.

Fig. 1 shows a pair of images with the correspondences superimposed. All the correspondences were used for the determination of the fundamental matrix. The correspondences on the skyline were used for the determination of the infinity homography. We obtained a RMS distance inferior to the pixel for the distance of points to corresponding epipolar lines, as well as for the distance of points to predicted correspondences by the infinity homography.

3 The profile lines

We first describe how, given a pair of affinely calibrated cameras, we can represent the profile lines. We then discuss a methodology for detecting these lines in a pair of images. Some very preliminary experimental results are presented.

3.1 An affine representation for fronto-parallel planes

Having performed affine calibration, if we know the vanishing line $\mathbf{r}$ of a plane in the first view, we can define a set of planes Π_Z which are parallel to this plane. No 3-D reconstruction is needed for that. Instead, the plane Π_Z is defined by its homography $\mathbf{H}_Z$, such that the projective coordinates of two points $\mathbf{m}$ and $\mathbf{m}'$, projections in the first and second image of a point of Π_Z, are related by:

$$\mathbf{m}' \simeq \mathbf{H}_Z\mathbf{m}$$

Let us consider the family of homographies:

$$\mathbf{H}_Z \simeq \mathbf{H}_\infty + \frac{1}{Z}\mathbf{e}'\mathbf{r}^T$$

where $\mathbf{e}'$ is the epipole in the second image. The direction of the plane Π_Z is given by its intersection L with the plane at infinity Π_∞, a line at infinity in 3D whose projection in the first image is the vanishing line of Π_Z in this image. Since the projections $\mathbf{m}$ of points of L satisfy $\mathbf{H}_Z\mathbf{m} \simeq \mathbf{H}_\infty\mathbf{m}$, the projective equation of the vanishing line is $\mathbf{r}^T\mathbf{m} = 0$. All the planes obtained by varying Z have the same vanishing line, therefore they are parallel. Note that such a construction could not have been done only in the projective framework where the notion of parallelism is not defined.

The representation of the terrain that we are interested in is a set of profile lines. Strictly speaking, a profile line is the trace in a vertical plane of the surface which represent the terrain. We can define a family of parallel vertical planes if the vertical vanishing point can be identified in the images. An example of such an identification in uncalibrated imagery is in [Reid and Zisserman, 1996]. If the vertical direction cannot be identified reliably, then we can still apply these ideas using the family of planes which are parallel to one of the camera's retinal plane (ie fronto-parallel with respect to this camera).

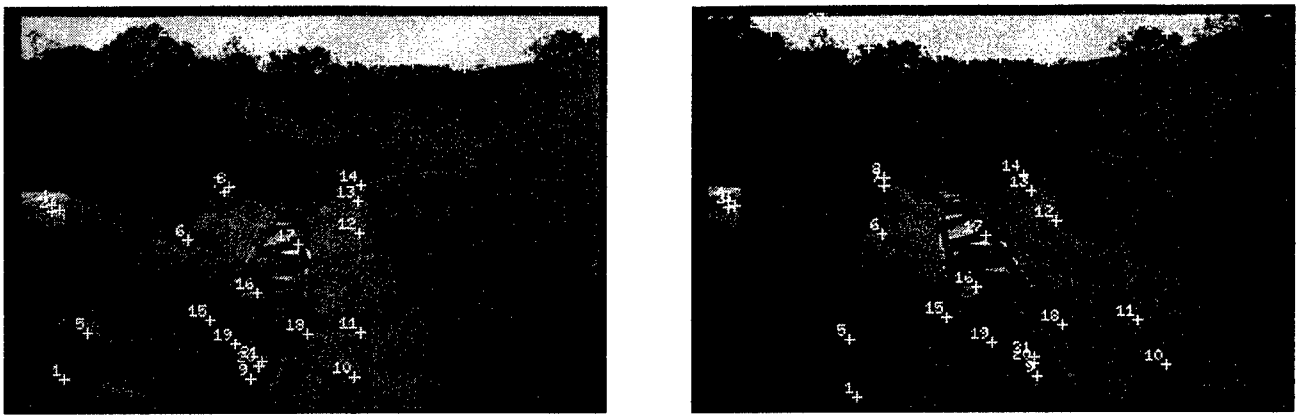


Figure 1: A pair of images with the correspondences used for affine calibration superimposed.

The particular case of the construction described above which is of interest to us here is fronto-parallel planes: planes parallel to the image plane of one of the cameras (which will be assumed to be the first camera). They are obtained with $\mathbf{r} = [0, 0, 1]^T$, which means that their vanishing line is the line at infinity in the image plane, ensuring that the image plane and the planes Π_Z are parallel. Each of these planes Π_Z is defined by its homography matrix:

$$\mathbf{H}_Z \simeq \mathbf{H}_\infty + \mathbf{e}'[0, 0, \frac{1}{Z}] = \begin{bmatrix} H_{11} & H_{12} & H_{13} + \frac{1}{Z}e'_1 \\ H_{21} & H_{22} & H_{23} + \frac{1}{Z}e'_2 \\ H_{31} & H_{32} & H_{33} + \frac{1}{Z}e'_3 \end{bmatrix}$$

A remarkable property is that although the calibration is only affine, the parameter Z has a metric interpretation. It represents the perpendicular distance of the plane to the origin, up to an unknown scale factor, Π_∞ being the plane at infinity, and Π_0 being the focal plane of the first camera. If $\mathbf{A}$ is the matrix of intrinsic parameters of the first camera, then the equation of the plane Π_Z in the coordinate system of the first camera is given by: $\mathbf{n}^T\mathbf{M} = d$, where $\mathbf{n}$, the unit vector normal to the plane and d the perpendicular distance of the plane to the origin, are given by:

$$\frac{\mathbf{n}}{d} = \mathbf{A}^T \begin{bmatrix} 0 \\ 0 \\ \frac{1}{Z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{Z} \end{bmatrix}$$

However, the epipole $\mathbf{e}'$ is determined only up to a scale factor, and this is why it is possible only to specify the depth of the plane Π_Z up to a scale factor.

3.2 Locating the profile lines

Charting a plane cross-section Knowing the homography $\mathbf{H}_Z$ makes it possible to determine whether a point $\mathbf{m}$ in the first image is the projection of a 3D point which belongs to the plane Π_Z: if it is the case, its correspondent in the second image should be $\mathbf{H}_Z\mathbf{m}$.

The idea is to compute a correlation score between the point $\mathbf{m}$ in the first image and the point $\mathbf{H}_Z\mathbf{m}$ in the second image. If $\mathbf{m}$ is a projection of a point which lies on the plane Π_Z, then this correlation score should be high. By computing such a score for each point of the first image, we create a correlation image, in which we expect the high values to correspond to points of the profile line.

From a computational point of view, for traditional correlation-based stereo, the total number of correlation scores to be computed is the image size times the number of pixels of the disparity interval. This is in addition to the rectification required to map the images into a pair with epipolar lines coincident to scan lines. In the approach that we propose, which does not require rectification, this number is at most the the image size times the number of depth samples. However, since some of the points $\mathbf{H}_Z\mathbf{m}$ will lie outside of the borders of the second image, and since the search zone is vertically reduced as the algorithm proceeds, as described latter, the number of correlation scores is significantly less than the upper bound.

In preliminary experiments, we have tried the

sum of squares of differences (SSD) and the cross-correlation (CC) scores, both in zero-mean (Z) and normalized mode (N), with various window sizes. It was found that the ZNCC score generated more false high correlation areas than the ZNSSD approach. An example of thresholded correlation image is shown in Fig. 2. It can be seen that the target profile line which goes through the white cylinder on the left is correctly located. However, there are other false high correlation areas. In this example, many of them can be eliminated just by increasing the size of the correlation window, however the horizontally repeating pattern at the base of the trees causes false high scores which cannot be eliminated. In order to find out whether this problem is due to the fact that we do not perform a search (as opposed to the traditional stereo, which underlies the approach of [Deriche *et al.*, 1996], and would negate the computationnal advantages provided by our approach), we applied the inverse homography $\mathbf{H}_Z^{-1}$ to the second image, which results in a pair of rectified images, and then applied a classical stereo correlation algorithm with back-validation. In Fig. 3 we show the sign of the disparity which was found. It can be seen that although the target profile lines actually separates the positive disparities from the negative disparities, there are still problems around the base of the trees. This suggests that the difficulty lies with defining a suitable correlation score, and we are in the process of investigating alternative approaches.

Taking into account the local orientation The classical measures assume that the neighborhood of a point can be locally approximated by a fronto-parallel plane, so that the transformation between a neighborhood in the first image and a neighborhood in the second image is only a translation, followed by a scalar transformation to take into account possible differences of photometric properties of the two cameras. This has proved to be very effective on aerial imagery, but is more questionable for ground-level images. A more general approach is to consider that the neighborhoods are local planar surfaces of arbitrary orientation. It has indeed been shown by Devernay and Faugeras [Devernay and Faugeras, 1994] that it is possible to recover the local orientation of the surface from correlation. However, this approach is computationally quite expensive, since it involves non-linear minimization at each pixel value. To avoid having to do such a computation, a possible approach is to precompute the local orientation of the plane at depth Z_i based on the orientations which can be deduced from the reconstructions at depth Z_{i-1} and Z_{i-2}. This is only an approximation, but in general it is much better than considering the local plane to be fronto-parallel.

Computation by curve evolution Using the correlation score described above, to locate a given profile line at depth Z is equivalent to find, by looking only into the first image, a curve which links all the points which maximize the correlation score. While several algorithmic approaches are possible, the methods of curve and surface evolution which have been developed in computer vision under the name of snakes [Kass *et al.*, 1988] and then reformulated by Caselles, Kimmel and Sapiro [Caselles *et al.*, 1995] and Kichenassamy et al. [Kichenassamy *et al.*, 1995] in the context of PDE driven evolving curves, are particularly suitable for solving the problem of locating the profile line.

The profile line is modeled as a deformable curve which optimizes two criteria: an objective function which represents the sum of the correlation scores over each point of the profile line, and a regularization term, which enforces regularity properties. This makes it possible to draw continuous profile lines which are reasonably smooth even if the correlation score is not very reliable.

The $2\frac{1}{2}$ hypothesis In the general case, the cross-sections by a plane are sets of closed curves, rather than a single curve. An elegant way to handle these topological difficulties is provided by the level set methods [Osher and Sethian, 1988, Sethian, 1990, Sethian, 1995]. However, in our case there is a major simplification, which makes it possible to consider a

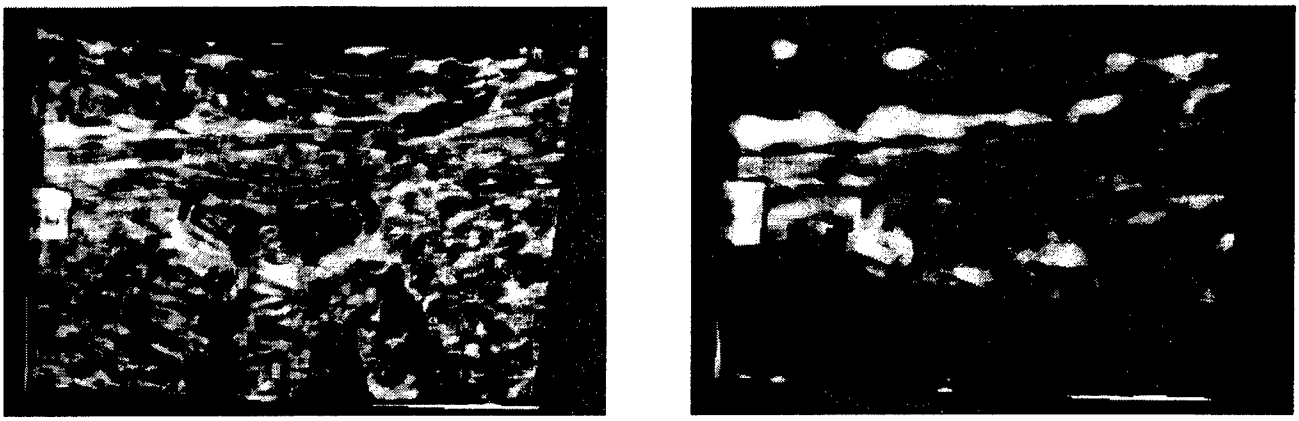


Figure 2: The correlation score between points $\mathbf{m}$ in the first image and points $\mathbf{H}_Z\mathbf{m}$, with window sizes of 9 and 19 pixels

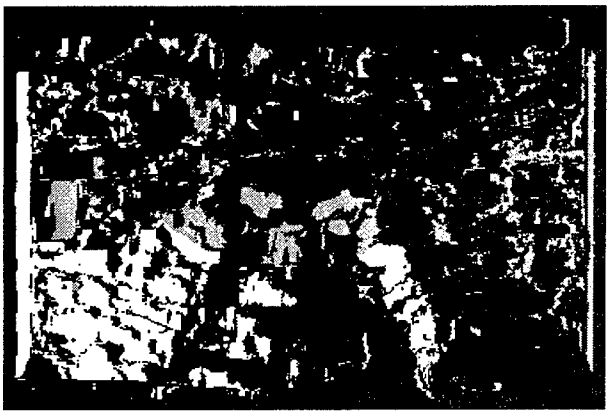


Figure 3: The sign of the disparity of the pair of images rectified with respect to the homography $\mathbf{H}_Z$. The profile lines separates the positive disparities from the negative disparities

profile line as a function $v = f(u)$, where u is the horizontal axis of the image and v the vertical axis. We assume that to each point (X,Y) is associated a single elevation $h(X, Y)$. This hypothesis is verified if there is no overhanging objects. In practice, although it is not satisfied by objects such as trees, however, since they have a rather small spatial extension, we shall consider when there are multiple elevations associated to a point (X,Y) (in the case of a tree: the ground, the lower branches, the upper branches), only the upper point. As it will be seen next, this is equivalent to consider that everything which is under an overhanging object is actually part of this object, an approximation which is generally acceptable for the tasks to be performed with the sketch. Another reason why such an approximation is reasonable is that because of the imprecisions associated with affine calibration and correlation, a profile line cannot be anyway located with an very high accuracy.

Taking into account ordering and occlusions Within the previous framework, using the fact that (a) the planes are ordered in depth, (b) anything under a profile line is considered to be solid, it is possible to take care of occlusions with a simple reasoning. We first determine the profile line which is closest to the cameras. This profile line should be limited only by the image frame. The next profile line is then above the previous one (in the image), and should be limited only by the image frame, and the previous profile at the bottom, and so on. By proceeding using the depth ordering of the parallel planes, we can make sure that the profile lines which are found do not include hidden parts.

Generating successive profile lines starting from the bottom of the image has two other benefits. First, when looking for the profile line at depth Z_i, we have to consider only the portion of the first image which lies above the profile line at depth Z_{i-1}. This reduces the amount of computations to be done. Second, we can use as a starting point for the evolving curve the profile line at depth Z_{i-1}, and let this curve propagate upwards in the image.

4 Summary

We have proposed a scheme for sketching natural terrain. This scheme takes advantage of general domain-specific constraints: the availability of an horizon line, and the $2\frac{1}{2}$ nature of the natural terrain. By taking advantage of these constraints, we are able to propose a method which has the potential to produce a useful representation from minimal data (two uncalibrated images) in a domain which has been traditionally considered to be difficult.

This method produces a dense sketch consisting of a set of profile lines where the order with respect to the dimensions of height above the ground plane and depth are correct. These profile lines are a semantically meaningful representation of natural terrain.

References

[Boufama *et al.*, 1993] B. Boufama, R. Mohr, and F. Veillon. Euclidean constraints for uncalibrated reconstruction. In *Proc. International Conference on Computer Vision*, pages 466–470, Berlin, Germany, 1993.

[Caselles *et al.*, 1995] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proceedings of the 5th International Conference on Computer Vision* [1995], pages 694–699.

[Collins, 1996] R.T. Collins. A space-sweep approach to true multi-image matching. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 358–363, San Francisco, CA, 1996.

[Deriche *et al.*, 1996] Rachid Deriche, Stéphane Bouvin, and Olivier Faugeras. A level-set approach for stereo. In *Fisrt Annual Symposium on Enabling Technologies for Law Enforcement and Security - SPIE Conference 2942 : Investigative Image Processing.*, Boston, Massachusetts USA, November 1996.

[Devernay and Faugeras, 1994] F. Devernay and O.D. Faugeras. Computing differential properties of 3-D shapes from stereoscopic images without 3-D models. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 208–213, Seattle, WA, 1994.

[Faugeras and Keriven, 1996] Olivier Faugeras and Renaud Keriven. Variational principles, surface evolution, pde's, level set methods and the stereo problem. Technical Report 3021, INRIA, November 1996.

[Faugeras *et al.*, 1995] Olivier Faugeras, Stéphane Laveau, Luc Robert, Cyril Zeller, and Gabriella Csurka. 3-d reconstruction of urban scenes from sequences of images. In A. Gruen, O. Kuebler, and P. Agouris, editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pages 145–168, Ascona, Switzerland, April 1995. ETH, Birkhauser Verlag. also INRIA Technical Report 2572.

[Faugeras, 1992] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. In *Proc. European Conference on Computer Vision*, pages 563–578, 1992.

[Fischler, 1996] M.A. Fischler. Robotic vision: Sketching natural scenes. In *ARPA Image Understanding Workshop*, Palm Springs, CA, 1996.

[Hartley *et al.*, 1992] R.I. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 761–764, Urbana, 1992.

[Hartley, 1994] R.I. Hartley. An algorithm for self calibration from several views. In *Proc. Conference on Computer Vision and Pattern*

Recognition, pages 908–912, Seattle, WA, 1994.

[icc, 1995] Boston, MA, June 1995. IEEE Computer Society Press.

[Kass *et al.*, 1988] M. Kass, A. Witkin, and D. Terzopoulos. SNAKES: Active contour models. *The International Journal of Computer Vision*, 1:321–332, January 1988.

[Kichenassamy *et al.*, 1995] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proc. Fifth International Conference on Computer Vision* [1995].

[Luong and Faugeras, 1996] Q.-T. Luong and O.D. Faugeras. The fundamental matrix: theory, algorithms, and stability analysis. *Intl. Journal of Computer Vision*, 17(1):43–76, 1996.

[Luong and Faugeras, 1997] Q.-T. Luong and O.D. Faugeras. Self calibration of a moving camera from point correspondences and fundamental matrices. *Intl. Journal of Computer Vision*, 22(3), 1997.

[Luong and Viéville, 1996] Q.-T. Luong and T. Viéville. Canonical representations for the geometries of multiple projective views. *Computer Visio and Image Understanding*, 64(2):193–229, 1996.

[Osher and Sethian, 1988] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed : algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[Quan, 1993] L. Quan. Affine stereo calibration for relative affine shape reconstruction. In *Proc. British Machine Vision Conference*, pages 659–668, 1993.

[Reid and Zisserman, 1996] I. Reid and A. Zisserman. Goal-directed video metrology. In *Proc. European Conference on Computer Vision*, pages II–645–658, Cambridge, UK, 1996.

[Robert and Deriche, 1996] L. Robert and R. Deriche. Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities. In Bernard Buxton, editor, *Proceedings of the 4th European Conference on Computer Vision*, Cambridge, UK, April 1996.

[Robert and Faugeras, 1993] L. Robert and O.D. Faugeras. Relative 3D Positioning and 3D Convex Hull Computation from a Weakly Calibrated Stereo Pair. In *Proc. International Conference on Computer Vision*, pages 540–543, Berlin, Germany, May 1993.

[Robert *et al.*, 1992] L. Robert, R. Deriche, and O.D. Faugeras. Dense depth recovery from stereo images. In *Proceedings of the European Conference on Artificial Intelligence*, pages 821–823, Vienna, Austria, August 1992.

[Sethian, 1990] J.A. Sethian. Numerical algorithms for propagating interfaces: Hamilton-jacobi equations and conservation laws. *Journal of Differential Geometry*, 31:131–161, 1990.

[Sethian, 1995] J.A. Sethian. Theory, algorithms, and applications of level set methods for propagating interfaces. Technical Report PAM-651, Center for Pure and Applied Mathematics, University of California, Berkeley, August 1995. To appear Acta Numerica.

[Shashua and Navab, 1994] A. Shashua and N. Navab. Relative affine structure: Theory and application to 3D reconstruction from perspective views. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 483–489, Seattle, WA, 1994.

[Zhang *et al.*, 1995] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 78:87–119, Oct 1995.

Design of Self-Tuning IU Systems

†Chandra Shekhar †Philippe Burlina ‡Sabine Moisan

† Center for Automation Research
University of Maryland
College Park, MD 20742

‡ INRIA Sophia-Antipolis
2004 Route des Lucioles
Sophia-Antipolis Cedex, France

Abstract

We propose a methodology for the development of image understanding systems that provide both convenience and flexibility. In this methodology, the image analyst provides the input data, specifies the IU task to be performed, and then provides feedback in the form of qualitative evaluations of the result(s) obtained. These assessments are interpreted in a knowledge-based framework to select the best algorithms and to find the most suitable parameter settings. In this manner the IU system is given the capacity to tune itself for optimal performance. A sample application (vehicle detection in aerial imagery) is developed to illustrate the approach.

1 Introduction

Image Understanding (IU) systems used in challenging operational environments should satisfy the conflicting requirements of flexibility and convenience. Flexibility is the ability of the system to accommodate variations in the characteristics of the input data. Convenience means that the system can be operated by an image analyst (IA) who is not familiar with the technical details of the algorithms employed.

Variations in image characteristics are caused by a number of factors such as weather, lighting conditions and image acquisition parameters. An IU system should accommodate a reasonable amount of such variations, and should degrade gracefully as the image characteristics deviate from the ideal. Modern IU systems allow for such variations by providing alternative algorithms for each task, as well as tuning parameters for each algorithm. In most cases, a judicious choice of algorithms and parameters provides results of acceptable quality under a wide range of operating conditions.

Most IU tasks, especially in defense applications, are handled by IAs who, while competent in the visual analysis of images, may not be familiar with the technical details of the algorithms they employ. It is not reasonable to expect the IA functioning in an operational situation to select and tune the algorithms for the task (s)he is required to perform. This function is best left to the designer of the system (the IU specialist) who may not be available during its operation. It is thus obvious that an IU system that provides flexibility in the choice of algorithms and parameter values may not be very convenient for the IA to utilize.

In order to achieve the conflicting goals of flexibility and convenience, we propose a framework to partially or fully automate the reasoning employed by the IU specialist in obtaining satisfactory results from the system. The algorithms in the system are semantically integrated into this framework, and the integrated system can be made available to the IA. A simplified model of a self-tuning IU system is shown in Fig. 1. This type of system is capable of self-tuning, i.e. adapting to changes in data characteristics and performance requirements with minimal external intervention. Any interaction with the IA is in terms of qualitative result evaluations, and not in terms of algorithms and parameters. In many situations, the same processing task is performed on a large data set consisting of hundreds or even thousands of images. In such cases, the system can be interactively tuned on some representative images, and once satisfactory performance is achieved, can then be used with fixed settings for batch processing of the remaining images in the data set.

In a previous paper [Shekhar *et al.*, 1996], we discussed the knowledge-based semantic integration of IU algorithms using the OCAPI architecture

The support of the Defense Advanced Research Projects Agency and the Office of Naval Research under Grant N00014-95-1-0521 is gratefully acknowledged.

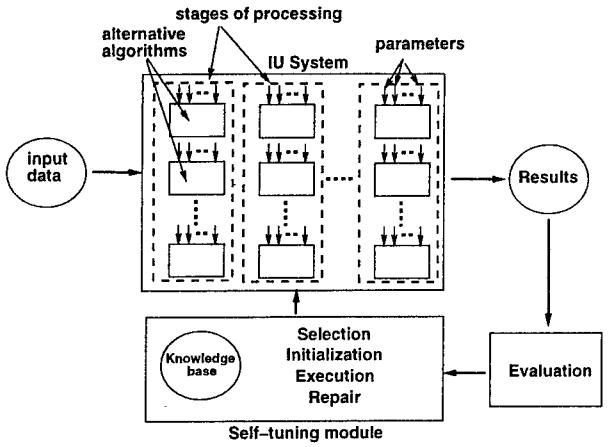


Figure 1: Architecture of a self-tuning IU system.

[Clément and Thonnat, 1993]. In this architecture, the reasoning of the IU specialist is formally represented using frames and production rules. Mechanisms are provided for program supervision tasks such as algorithm selection and tuning. In this paper, we extend this work to handle more complex program supervision strategies. We use the LAMA (Library for the Adaptive Management of Algorithms) architecture [Vincent *et al.*, 1996] to implement our ideas.

The organization of this paper is as follows. Section 2 briefly reviews related work. Section 3 discusses the basic concepts of a self-tuning IU system. Section 4 presents the LAMA architecture for developing adaptive applications. In Section 5, results of applying our methodology to the vehicle detection problem are presented. The final section contains the conclusions resulting from our work.

2 Review of previous work

Knowledge-based systems have been traditionally used for the high-level interpretation of images, and for specific IU tasks such as segmentation (e.g. [Nazif and Levine, 1984]). These systems incorporate mechanisms for the spatial and temporal reasoning that is characteristic of intermediate- and high-level image understanding. For an excellent survey of the various knowledge-based systems and techniques that have been developed in this context, see [Crevier and Lepage, 1997].

The use of a knowledge-based approach for the development of self-tuning IU systems is a relatively recent phenomenon [Clément and Thonnat, 1993; Crevier and Lepage, 1997]. Some of the early work is reported in [Hanson and Riseman, 1978; Toriu *et al.*, 1987; Matsuyama, 1989]. More recently, this problem has been addressed in the context of the VIDIMUS project [Bodington, 1995], with the aim of developing an intelligent IU environment for industrial inspection. The automatic generation of an image processing script based on a user request and a knowledge-based model of an application domain is addressed in [Chien, 1994]. In [Draper, 1996] IU algorithm control is posed as a Markov decision problem. In [Strat, 1993; Strat and Fischler, 1991] a context-based vision paradigm is proposed, where the basic aim is to use contextual information to select methods and parameters in an IU application. The use of contextual information derived from site models to construct control patches for the self-tuning of IU algorithms is discussed in [Burlina *et al.*, 1997].

3 Self-tuning

In a typical IU application, a number of stages of processing are involved in going from the raw input data to the final result, as shown in Fig. 1. Typically, at each stage of processing a number of alternative algorithms can be employed. Each of these algorithms, in turn, may have one or more tunable parameters. These parameters may be continuously variable, or may take discrete set of values. Self-tuning is the ability of the IU system to select the appropriate algorithms and parameter values based on the input data, contextual information, and result evaluations. In most cases, the selection of algorithms may be performed, before any data processing has taken place, based on the contextual information alone [Strat, 1993]. Parameter tuning, on the other hand, is dependent on the characteristics of the specific data set. It can only be performed in real time, interleaved with the processing of the data, based on the operator's evaluation of the results. In this paper we examine the parameter tuning problem in detail.

3.1 Ideal operating point

Consider an IU system A composed of m stages, with n_i parameters in the ith stage:

$$A = A_1 A_2 \ldots A_i \ldots A_m$$

where each A_i is of the form

$$A_i = A_i(p_{i1}, p_{i2}, \ldots, p_{in_i})$$

As shown in Fig. 2 the entire IU system can be regarded as consisting of a single algorithm $A(p_1, p_2, \ldots, p_N)$. The N parameters that tune this "black box" algorithm are the $\sum_{i=1}^{m} n_i$ parameters of the m individual stages. We refer to a specific setting of these N parameters as an operating point

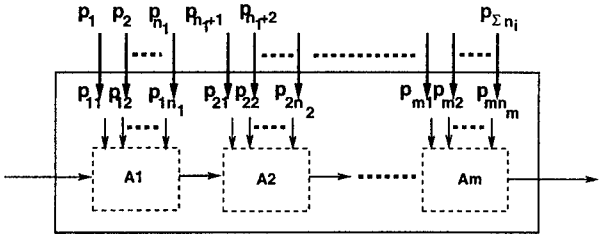


Figure 2: A simplified black-box model of an IU system consisting of stages A_1–A_m. The parameters of the individual modules become the parameters of the black box.

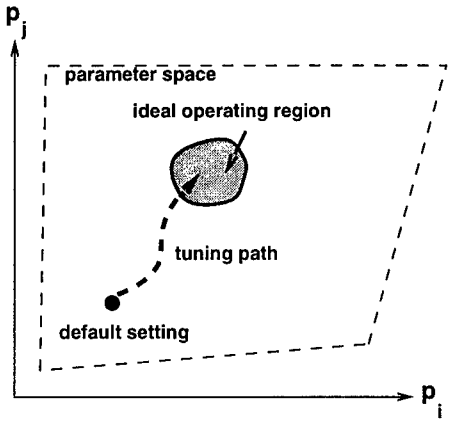


Figure 3: Parameter tuning viewed as a search for an ideal operating point in the parameter space.

(OP). We define an *ideal* operating point as a parameter seting that yields satisfactory performance for the given data set. Our basic assumption is that there exists at least one ideal operating point (or, in the case of continuous-valued parameters, operating region). In general, the default parameter setting will not be an ideal one. The objective, then, is to be able to find an ideal OP for the given data set in an efficient manner. If each parameter p_i has k_i possible values, the total number of parameter settings is $\prod_i k_i$. An exhaustive search of the N-dimensional parameter space may therefore be too computationally expensive. This is where a knowledge-based approach is applicable. An IU specialist employs a problem-solving strategy consisting of heuristics, rules of thumb, etc., which effectively help him or her find a short cut ("tuning path") to an ideal operating point, as shown in Fig. 3, without having to explore the entire parameter space. The strategy employed by the specialist has to be integrated into the IU system to give it a self-tuning capability.

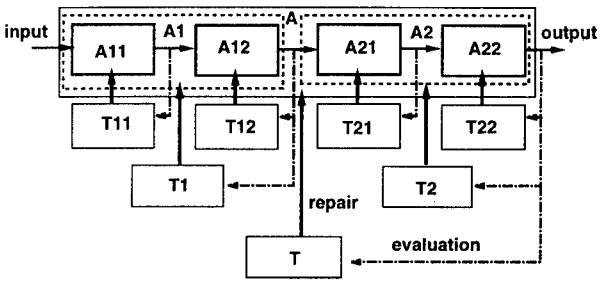


Figure 4: Specialist tuning mode for a three-level IU system. The results of the modules A, A_1, A_2 and A_{11}–A_{22} and the corresponding strategies T, T_1, T_2 and T_{11}–T_{22} are available.

3.2 Modes of self-tuning

Any IU task A can be hierarchically decomposed into a set of subtasks A_1–A_m, each A_i of which may be decomposed further into subtasks A_{i1}–A_{im_i} and so on. For instance, a typical IU system may consist of a top-level module A, consisting of sub-modules A_1 and A_2, and these may be composed of elementary subtasks A_{11}, A_{12}, A_{21}, and A_{22}. Attached to each (sub)task $A_{...}$ at any level in the hierarchy is a *strategy* $T_{...}$ which contains all the specialist's knowledge about the module: how/when it should be used, how to evaluate its performance, and how to adjust it if improved performance is required. Depending on the strategies available, a self-tuning IU system can function in one of two modes: the specialist mode or the user mode.

In the specialist mode, shown in Fig. 4, all the strategies at every level of the hierarchy are available. In other words, results at every stage, including the intermediate ones, are available for evaluation by the specialist. This is applicable to the test phase when the specialist is in the process of testing the functioning of the system. In the user mode, shown in Fig. 5, only the top-level strategy T is directly available and only the results of the final stage are available for evaluation by the IA. This mode is designed for the operational phase. From a control-theoretic viewpoint, a self-tuning IU system can be regarded as a closed-loop system where the observer is the IA, who provides feedback in the form of result evaluations. This feedback can be either at the highest level (corresponding to the user mode), or at all levels (corresponding to the specialist mode).

In the specialist mode, the availability of intermediate results enables linear planning for the fine-grain local optimization of algorithms. The user mode,

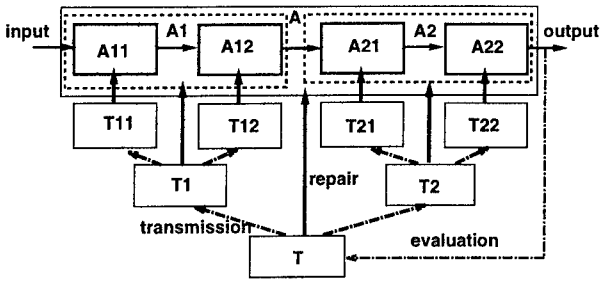


Figure 5: User tuning mode for a three-level IU system. Only the results of the top-level module A and the corresponding strategy T are directly available. The strategies T_1, T_2 and T_{11}–T_{22} are available only indirectly through message transmission.

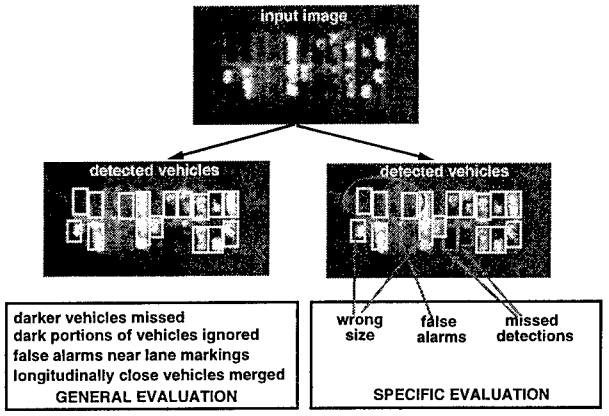


Figure 6: Example of the two types of result evaluation. The application here is the detection of vehicles in an aerial image from the TEC-2 data set.

however, necessitates the use of more complex reasoning, since only the final result is available, based on which any algorithm at any stage of the processing may have to be retuned. Previous work, reported in [Shekhar *et al.*, 1996], dealt mainly with the specialist mode. In the present work, the user mode is the primary focus.

3.3 Result evaluation

In the user mode, the IU system is tuned based on the evaluation of results by the IA. We can define two types of result evaluation: general and specific. General evaluation consists of global judgments about algorithms and results ("too many false alarms", "too many missed detections at road intersections", etc.). Specific evaluation, on the other hand, pertains to particular objects or regions in the result ("this airplane is a false alarm", "this portion of the image contains too many false alarms", etc.). This is illustrated in Fig. 6. In our present work, we deal exclusively with general evaluations. Specific evaluations, although more detailed and informative, are also difficult to deal with in a general-purpose framework. They require mechanisms for reasoning and for user interaction that are application-dependent. This makes separation of the application from the knowledge base more difficult. Finally, specific evaluations are not very useful for batch processing.

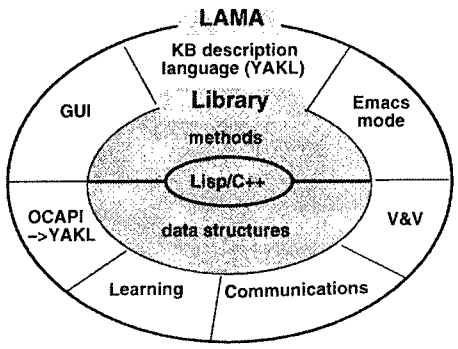


Figure 7: The LAMA platform

4 The LAMA platform

LAMA is a methodology as well as a general-purpose platform for developing intelligent applications, consisting of a kernel library, a knowledge-base description language YAKL (Yet Another Knowledge-base Language), verification and validation (V & V) facilities, a graphical user interface and other tools, as shown in Fig. 7. A complete description of LAMA is beyond the scope of this paper; the interested reader is referred to [Vincent *et al.*, 1996]. For our purposes, LAMA may be viewed as an architecture based on frames and rules for encapsulating the problem-solving knowledge of the IU specialist.

A self-tuning IU application developed using LAMA consists of a set of pre-existing algorithms (also referred to as programs, modules or methods), a knowledge base (KB) on using these algorithms, and a control (supervision) engine. Knowledge about IU algorithms is expressed at two levels of abstraction. A *goal* is the abstract form of an IU functionality, which is realized in a more concrete form by one or more *operators* corresponding to it. An operator may be either simple, corresponding to an executable program, or composite, represented by a predefined skeletal plan. A skeletal plan describes a network of connections between operators (choice, sequence, repetition, etc.) for achieving a given goal.

The description of an operator contains information about its arguments (name, type, range, default values, etc. of the input data, output data and parameters), semantic information about its applicability (in the form of pre- and post-conditions), as well as criteria for parameter initialization, result evaluation, etc. For operators corresponding to real executable programs the calling syntax is also provided.

The functioning of the control engine can be decomposed into a number of phases: *planning* and *execution* of programs, *evaluation* of the results, and *repair*. The planning step first builds a plan, or part of a plan, which is then executed. The results of execution are then assessed in the evaluation step. The evaluation mechanism consists of a collection of *assessment* rules which can be used to evaluate an algorithm output, a parameter or an algorithm in its entirety. If the assessments are positive, the planning process continues. If failures are detected, the repair step invokes the appropriate remedial measures, encoded in the form of *repair* rules, which may either result in re-execution or in re-planning. Failure handling is performed either locally inside the operator, or nonlocally by message transmission to another part of the plan. Local failures are handled by parameter tuning (specified using *adjustment* rules).

4.1 Integrating an application into LAMA

IU applications typically consist of a number of executable programs with associated syntactic and semantic knowledge. Syntactic knowledge consists of calling syntax, data formats, etc. Semantic knowledge is the specialist's expertise about the use of the programs. An application is integrated into LAMA by encoding both types of knowledge using the structures described in the previous section. A knowledge representation language, YAKL, is used for this purpose. Mechanisms are provided to test the consistency of the knowledge base thus created. A number of different control engines are available for running the integrated application. A GUI is provided for examining the knowledge base as well as for monitoring the application during execution.

5 Example: Vehicle detection

The methodology discussed in Section 4 has been applied to the detection of vehicles in aerial imagery. We have used a simplified version of the vehicle detector developed at the University of Maryland [Chellappa *et al.*, 1996]. The objective is to detect and approximately localize vehicles of a specified size and orientation. In the current implementation, a precise localization and geometric description of the detected vehicles is not attempted, nor do we consider vehicles of differing sizes and orientations. The main stages of processing, shown in Fig. 8, are as follows.

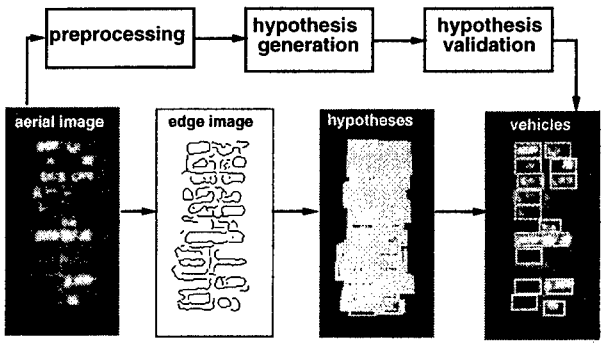


Figure 8: A simplified version of the UMD vehicle detector.

Preprocessing: Edge pixels are extracted using the Canny edge detector. Both gradient magnitude and gradient direction are computed.

Hypothesis generation: A modified generalized Hough transform (GHT) is used to locate areas corresponding to centers of candidate vehicles. Edge pixels vote for all possible centers of vehicle contours which contain the pixel. The votes are collected in an accumulator array, and thresholded. The result is a set of hypothesized vehicle centers. Local "rubber-band" contour matching is subsequently applied to reject candidate vehicles which do not have sufficient support boundaries on both sides of the vehicle. The rubber-band matching technique ensures unique lateral counts of edge pixels within the template bandwidth.

Hypothesis verification: This stage resolves spatial conflicts (overlaps) between vehicle hypotheses. This is done in three steps. In the first step, the conflict resolution is done purely on the basis of distances between the centers of candidate vehicles. If two candidate vehicles are closer than a certain fraction of their width, the one with the greater boundary support is retained. The second step uses the size of the overlapping area between two conflicting vehicles as the criterion to reject the weaker vehicle. In the final step, the longitudinal distance between adjacent vehicles lying on the same axis is used as the filtering criterion.

Vehicle grouping: This optional step uses inter-vehicle spacing as a basis for grouping vehicles into clusters.

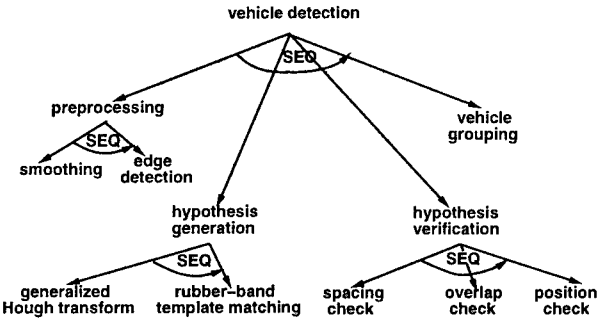


Figure 9: Operator hierarchy for the UMD vehicle detector.

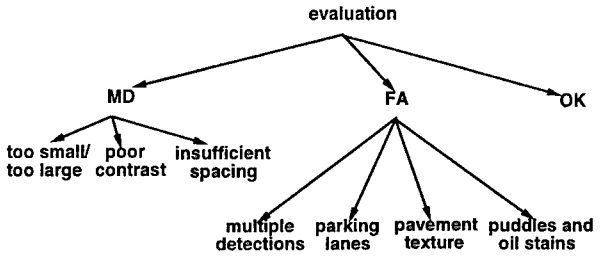


Figure 10: User evaluation of the UMD vehicle detector.

5.1 Knowledge base

The operator hierarchy for the vehicle detector is shown in Fig. 9. The knowledge base is under development. Currently, it consists of 12 operators (4 composite and 8 simple), 9 sequential operator links, and 22 rules (6 repair, 2 initialization, 10 assessment and 4 adjustment). As the knowledge base for this application is developed further, the number of operators is expected to increase only slightly, whereas the number of rules is expected to increase considerably.

5.1.1 Evaluation and repair strategies

As in any target detection system, there are two principal types of errors: missed detections (MDs) and false alarms (FAs). The general objective is to reduce both types of errors as much as possible. In practice, some tradeoff is made between the MD rate and the FA rate. Currently, the user is asked to choose between the responses MD (too many missed detections), FA (too many false alarms), and OK (results are satisfactory). A future version will deal with the common situation where errors of both kinds are simultaneously present. If the response is not OK, then the user is further queried about the

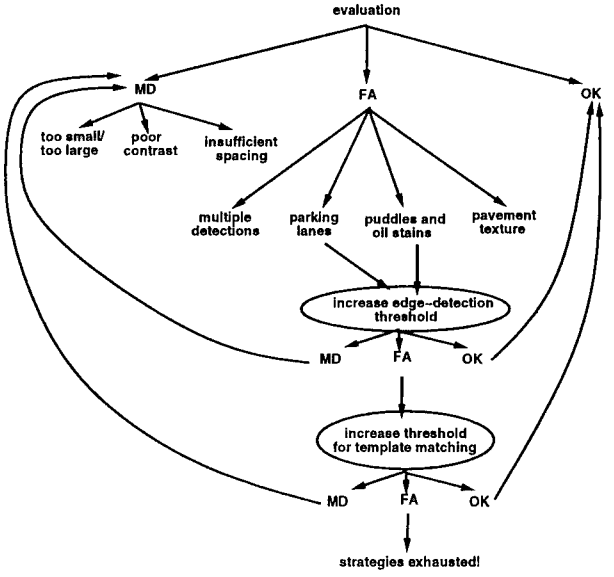


Figure 11: Example of repair strategy.

type of MD or FA, as shown in Fig. 10. Currently, MDs due to the following three situations are recognized: vehicles too large or too small, vehicles with low contrast, vehicles too tightly packed. Four types of FAs are handled: multiple hits from the same vehicles, false positives at parking lanes, puddles/oil stains mistaken for vehicles, FAs due to pavement texture. Extensive testing on a diverse set of aerial images will enable us to create a richer taxonomy of errors.

The repair mechanism has a nested structure, and is interleaved with the evaluation mechanism. For every allowed error subtype there are one or more repair strategies. The repair strategies are tried one after the other until either the error disappears or the strategies are exhausted. An example is shown in Fig. 11, and sample results in Fig. 12.

An alternative approach to this problem is taken in [Burlina *et al.*, 1997], where it is posed as the determination of the optimal operating point on a Receiver Operating Point (ROC) curve [Poor, 1988]. It is then solved using an optimization method. This is applicable when there is a single tuning parameter (generally some kind of threshold).

6 Conclusions and future work

This paper has presented a methodology for adding flexibility and convenience to an existing IU system by integrating the IU specialist's knowledge into it using a knowledge-based architecture. The system can then self-tune in response to the IA's evaluations.

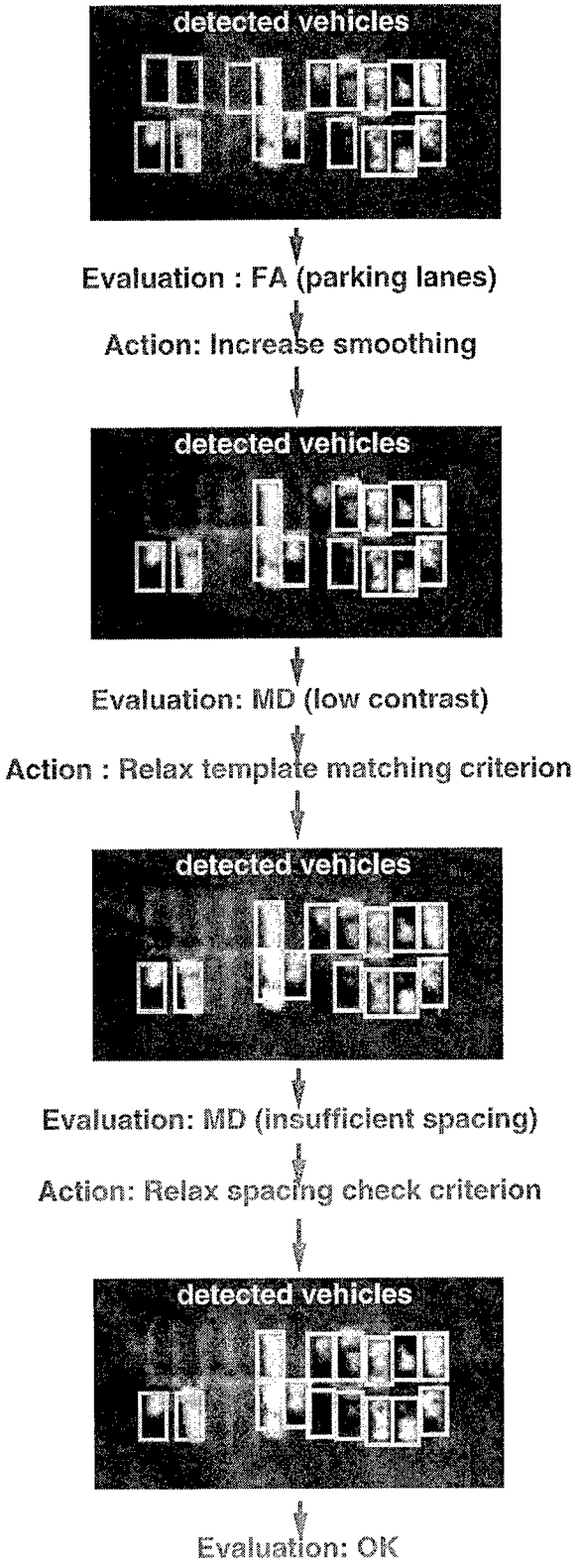


Figure 12: Sample results of evaluation and repair.

The proposed methodology assumes that the IU system has a non-empty operating region at which it yields satisfactory results. It imitates the strategy of the specialist in reaching a point in this region from a given or default setting. Obviously, if no combination of the tuning parameters can yield satisfactory results, neither the specialist nor the self-tuning framework will have any possibility of succeeding. On the other hand, if the self-tuning strategy does not capture the full complexity of the specialist's reasoning, it may fail in difficult cases even if a solution exists. Detecting this failure may not always be easy for the IA, since the self-tuning strategy may have loops and other complex chains of reasoning. If the system does not solve the problem in a reasonable amount of time, it should be considered as having failed. Experimentation on large and diverse data sets and constant refinement of the knowledge base will ensure that such failures do not occur too often.

Currently, result evaluations are in the form of general remarks about the results obtained, and not about specific portions or objects of the output. Our future work will incorporate some mechanisms for handling specific evaluations. We will also consider a more sophisticated version of the vehicle detector capable of detecting vehicles of all sizes and orientations. We also propose to test the methodology on other candidate problems such as multisensor registration.

Acknowledgments

We are grateful to Vasudev Parameswaran, Regis Vincent and Monique Thonnat for their valuable contributions to various aspects of the work reported in this paper. We would also like to thank Profs. Rama Chellappa and Azriel Rosenfeld for their comments and suggestions.

References

[Bodington, 1995] R. Bodington. A software environment for the automatic configuration of inspection systems. In *First International Workshop on Knowledge-Based Systems for the (re)Use of Program Libraries*, INRIA, Sophia Antipolis, France, November 1995.

[Burlina *et al.*, 1997] P. Burlina, V. Parameswaran, and R. Chellappa. Sensitivity analysis and learning strategies for context-based vehicle detection algorithms. In these Proceedings.

[Chellappa *et al.*, 1996] R. Chellappa, X. Zhang, P. Burlina, C. L. Lin, Q. Zheng, L. S. Davis, and A. Rosenfeld. An integrated system for site-model supported monitoring of transportation activities in aerial images. In *DARPA Image Understanding Workshop*, pages 275–304, Palm Springs, CA, February 1996.

[Chien, 1994] S. A. Chien. Using AI planning techniques to automatically generate image processing procedures: A preliminary report. In *Second International Conference on AI Planning Systems*, pages 219–224, Chicago, IL, June 1994.

[Clément and Thonnat, 1993]
V. Clément and M. Thonnat. A knowledge-based approach to the integration of image processing procedures. *CVGIP: Image Understanding*, 57:166–184, 1993.

[Crevier and Lepage, 1997]
D. Crevier and R. Lepage. Knowledge-based image understanding systems: A survey. *Computer Vision and Image Understanding*, 1997. In press.

[Draper, 1996] B. Draper. Modeling object recognition as a Markov decision process. In *IAPR International Conference on Pattern Recognition*, volume 4, pages 95–99, Vienna, Austria, August 1996.

[Hanson and Riseman, 1978] A. R. Hanson and E. M. Riseman. VISIONS: A computer system for interpreting scenes. In A. Hanson and E. Riseman, editors, *Computer Vision Systems*. Academic Press, San Francisco, CA, 1978.

[Matsuyama, 1989] T. Matsuyama. Expert systems for image processing: Knowledge-based composition of image analysis processes. *Computer Vision, Graphics and Image Processing*, 48:22–49, 1989.

[Nazif and Levine, 1984] A. M. Nazif and M. D. Levine. Low-level image segmentation: An expert system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:555–577, 1984.

[Poor, 1988] V. Poor. *An Introduction to Signal Detection and Estimation*. Springer-Verlag, Berlin, 1988.

[Shekhar *et al.*, 1996] C. Shekhar, S. Kuttikkad, and R. Chellappa. Knowledge-based integration of IU algorithms. In *DARPA Image Understanding Workshop*, pages 1525–1532, Palm Springs, CA, February 1996.

[Strat, 1993] T. Strat. Employing contextual information in computer vision. In *DARPA Image Understanding Workshop*, pages 217–229, Washington, DC, April 1993.

[Strat and Fischler, 1991] T. Strat and M.A. Fischler. Context-based vision: Recognizing objects using both 2D and 3D imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:1050–1065, 1991.

[Toriu *et al.*, 1987]
T. Toriu, H. Iwase, and M Yoshida. An expert system for image processing. *Fujitsu Sci. Tech. Journal*, 23.2:111–118, 1987.

[Vincent *et al.*, 1996] R. Vincent, S. Moisan, and M. Thonnat. A library for program supervision engines. Technical Report 3011, I.N.R.I.A., Sophia Antipolis, France, 1996.

SUPER RESOLUTION WITH REGION SENSITIVE INTERPOLATION

Krishna Ratakonda and Narendra Ahuja
Department of Electrical and Computer Engineering
Beckman Institute, University of Illinois, Urbana, IL 61801.
krishna,ahuja@stereo.ai.uiuc.edu

Abstract

Super resolution aims to provide a magnified image, several times the size of a given small image, while avoiding blurring, ringing or other artifacts. Previous approaches to this problem produced either increased artifacts and noise or over smoothed images [8, 4, 11, 6]. We propose a new approach which relies on interpolating where it is justified and not interpolating across edges. Previous methods which selectively interpolate across edges [10, 2] tend to promote false edges leading to noticeable artifacts. This occurs due to the imprecise location of the edges in the magnified image (since we only have access to a sub-sampled image) and the algorithms make one-step decisions as to the course of action in edge-areas of the image. The adaptive nature of the proposed scheme is aimed at avoiding such an error by not committing blindly to a predetermined course of action at the edge locations. Further more, we avoid stair case (or smoothing) approximation to the edges in the magnified images (which causes annoying artifacts) by using a simple scheme which performs very well in practice. The multi-scale nature of the transform [1] may be utilized to further reduce artifacts due to false edges.

1 Introduction

Resolution enhancement involves the problem of magnifying a small image to several times its actual size while avoiding blurring, ringing or other artifacts. Classical methods include bilinear, bi-cubic or FIR interpolation schemes followed by a sharpening method like unsharp masking [8, 5]. Interpolation schemes tend to blur the images when applied indiscriminately. Unsharp masking, which involves subtracting a properly scaled Laplacian of the image from itself, produces artifacts and increases noise. More sophisticated schemes involving Wavelet or Fractal based techniques have also been proposed [4, 11, 6, 7]. Such methods perform extrapolation of the signal in either the Wavelet or Fractal domain, which lead to objectionable artifacts when the assumptions behind such extrapolation are violated. It may also be noted that such extrapolatory assumptions predict and actively enhance the high frequency content within the image thus increasing any noise present in the sub-sampled image. In this paper, we propose an iterative method which improves the performance of *any given base interpolation scheme* while not making explicit "high frequency enhancing" assumptions.

The key idea behind the proposed method is: *interpolation is good until we interpolate across an edge.* So, instead of making ad hoc extrapolatory assumptions, we depend on interpolating in the "right fashion". Methods which selectively interpolate across edges have been previously proposed [10, 2, 3]. Such methods tend to promote false edges leading to noticeable artifacts. This occurs because the location of the edges in the magnified image is itself imprecise (since we only have access to a sub-sampled image) and the algorithms make one-step decisions as to the course of action in edge-areas of the image. The iterative nature of the proposed scheme is aimed at avoiding such an error by not committing blindly to a predetermined course of action at edge locations. We allow the edge pixels to vary within a confidence interval of $(+\delta, -\delta)$ around the initial values assigned to them. The final values attained by the edge pixels within this confidence interval are determined by other constraints which we would like the reconstructed image to satisfy (please see following paragraph or Section 2). Further more, the multi-scale segmentation algorithm used to find edges [1] provides segmentation information (and hence edges) at different scales of gray level homogeneity. So, one can avoid "weak edges" that cause errors, by selecting the right scale of segmentation. How ever, this usually decreases the sharpness of the image. In this paper, we provide results at two different scales

*This research was supported by the Advanced Research Projects Agency under grant N00014-93-1-1167 administered by the Office of Naval Research and the NSF grant IRI 93-19038.

of segmentation (a fine scale and a coarse scale).

The question to answer is: how do we obtain an iterative algorithm which provides a good reconstructed image starting with the sub-sampled image and segmentation information from the algorithm in [1]? We formulate the iterative procedure within the projection on convex sets (POCS) formalism [9]. POCS is used to find a solution which lies at the intersection of various convex constraint sets. One of the convex sets has already been described in the previous paragraph. It involves constraining the edge pixels to lie within a confidence interval around the initial values assigned to them (please see Section 2 for more details). Another constraint that naturally arises is that the values at non-edge locations should lie within a confidence interval around the values generated by the base interpolation scheme. This forms the second convex constraint to be satisfied by the reconstructed image. It is well known that sub-sampling retains the low frequency content of the image in the Fourier domain. Thus constraining the low frequency content in the Fourier domain to remain unchanged in both the sub-sampled and the reconstructed image forms the third and final constraint (which is also a convex set). This kind of formulation has been treated by us in a general context in [12].

In order to facilitate comparison, we subsample the Lena image and then reconstruct the image using (a) the base interpolation scheme and (b) the proposed scheme. Results indicate distinct reduction in blurring and improved quality (both visually and in terms of the PSNR) in the reconstructed images for the proposed method over the base interpolation scheme. The comparison in each case is made with respect to the original, unsubsampled image. In two other sets of results we start with a small image (taken from the standard images Barbara and Gold hill and not obtained by sub-sampling) and then employ (a) the baseline scheme and (b) proposed scheme for magnification. Section 2 describes the proposed scheme in more detail while a few illustrative results are in Section 3.

2 Enhancement Scheme

The interpolation scheme proceeds in three steps: (a) obtain an interpolated image with the base interpolation scheme, (b) obtain segmentation information from the multi-scale segmentation algorithm and (c) use an iterative algorithm to reconstruct the image. We assume that we have a base interpolation scheme to start with [8]. The second step in the proposed method is to obtain an edge mask locating edges of interest in the image. There is more than one way to do this. [10, 2] find the edges from the sub-sampled image and then finds their approximate locations in the magnified image. This leads to a staircase (or smoothing) approximation of the edges and causes visual artifacts. We found that a better approach is to interpolate the image (with the base interpolation scheme) and then find the edges from the interpolated image. This scheme is based on the assumption that it is better to find edges directly in an interpolated image rather than find edges in the small image and then interpolate the edge locations. This assumption bears out well in practice and is much simpler to implement. Segmentation (and hence edges) are found using a recently proposed multi-scale segmentation algorithm [1].

2.1 Outline

As explained in the previous section, the reconstruction algorithm is constructed using the POCS formalism [9, 12]. In order to define the algorithm we need to define the convex constraint sets. The solution (reconstructed image) lies at the intersection of the following convex sets:

1. The values in the non-edge locations are constrained to vary within limits ($+\delta_1$,$-\delta_1$) from their interpolated value.
2. The values in the edge locations are constrained to vary within limits ($+\delta_2$,$-\delta_2$) from their predicted value. The predicted value is found by averaging over the nearest 8-pixel neighborhood with appropriate weighting corresponding to distance. A weight of zero is given to those pixels which do not lie in the same region as the current pixel.
3. In the Fourier domain, low frequency values are constrained to be the same as those obtained by taking the Fourier transform and scaling (by zero padding the DFT) the initial, unmagnified image.

The need for the first two constraints is evident (as described in the introduction). They constrain the solution to be close to the model generated by an appropriate combination of the segmentation and interpolation schemes within the confidence limits set by δ_1 and δ_2. The last constraint is obtained from the fact that a sub-sampled image in two dimensions preserves the low frequency content of the original image. Appropriate scaling of the frequency values is necessary in order to account for the size change due to magnification. For example, a 4X magnification means that we have (1/16)th of the Fourier coefficients from the sub-sampled image. In the absence of noise in the sub-sampled image and any apriori constraints on the enhanced image, we use all the available Fourier data.

2.2 Control parameters

In Section 2.1, three different control parameters were used i.e., δ_1, δ_2 and the scale of segmentation. The question which naturally arises is: how do we select these parameters? The answer is dependent on various criteria:

(1) the base interpolation scheme used effects the confidence interval $(+\delta_1, -\delta_1)$, (2) if edge sharpness is the primary criterion, $(+\delta_2, -\delta_2)$ should be small, and (3) if the magnification is large, the confidence in the edge locations is reduced and this should be reflected in choosing the δ values. For a fixed base interpolation scheme, similar edge sharpness criteria and the same magnification, a single set of δ values should give good results. This observation is verified in practice. This fixed set of δ's may be obtained by testing the fidelity of the initial iterate over a wide class of images. We can thus use the following scheme averaged over a training set of images:(1) sub-sample the image without aliasing, (2) magnify the sub-sampled image to the original size (using the scheme described in Section 2 which gives the initial value of the image for the iteration process) and (3) find the expected value of the absolute error at the edges (determines δ_1) and the interior of the regions (determines δ_2) separately. In order to generate the results shown in this paper which required 4X magnification and had bi-linear interpolation for the baseline scheme, we used fixed values of δ_1 and δ_2 which were 1.0 and 5.0 respectively. We do not claim that these choices of δ's are the best possible values in all applications. In some situations (for example medical images) it might be better to have sharper edges at the expense of artifacts. Practically, a fixed choice of δ's is found to give good results for a wide range of images, provided that the criteria which went into the selection of the values were not changed.

Selecting a particular scale of segmentation is aimed mainly at avoiding the weak and false edges generated by the baseline interpolation scheme which precedes segmentation. In the results shown we used two different segmentation scales, one at a fine level and the other at a coarse level. The fine scale of segmentation tends to enhance artifacts in some cases, but in general improves the sharpness of the image.

3 Results

Figure 1.1 gives the results of magnification at two different segmentation scales using the proposed method. For the results shown in this paper POCS required 3-4 iterations to converge to within 10% of the final solution (in terms of mean square error). The initial unmagnified image was obtained by sub-sampling the original image (Lena) by a factor of 4 (avoiding aliasing by low pass filtering). The base interpolation scheme used in this case was bilinear interpolation. At both scales of segmentation we get more than 1dB improvement in the PSNR for the region of interest shown in the figure. The PSNR is found by comparing the region of interest with original Lena image. It may be noted that some artifacts (especially around the nose region) may be found in figure 1.1(c). This occurs because the scale of segmentation chosen detected finer edges than necessary. Two other sets of results are presented in figure 1.2(a)-(c) and figure 1.3(a)-(c). These results were obtained by using small clips of standard images (not obtained from subsampling a larger image). The images were then magnified 4X using the baseline scheme and the proposed scheme. In figure 1.2 the striped shawl is enhanced with the stripes clearly visible as opposed to linear interpolation which blurs the stripes. Note that the face portion of figure 1.2 is too blurred to be enhanced very much and the algorithm adapts to this situation correctly without producing artifacts. In figure 1.3, the bars on the windows become more distinct and the texture on the wall becomes enhanced. The images show the efficacy of the method in scenes with different textures and varying amounts of detail.

References

[1] N. Ahuja. A transform for the detection of multiscale structure. *IEEE Trans. PAMI*, December 1996.

[2] J. P. Allebach and W. P. Wong. Edge directed interpolation. *IEEE Conference on Image Processing*, 3:707–11, September 1996.

[3] S. Carrato, G. Ramponi, et al. A simple edge-sensitive image interpolation filter. *ICIP*, 3:707–11, September 1996.

[4] S. G. Chang, Z. Cvetkovic, and M. Vetterli. Resolution enhancement of images using Wavelet transform extrema extrapolation. *Proceedings ICASSP*, 4:2379–82, May 1995.

[5] H. Chen and G. E. Ford. An fir image interpolation filter design method based on properties of human vision. *Proceedings ICIP*, 3:581–85, 1994.

[6] M. Gharavi-Alkhansari. Resolution enhancement of images using a Fractal model. *Ph. D. Thesis, University of Illinois at Urbana*, 1996.

[7] B. R. Hunt et al. Super resolution of imagery: Algorithms, principles and performance. *Intl. J. Imaging Systems and Tech.*, 6(4):297–308, 1995.

[8] A. Jain. Fundamentals of digital signal processing. *Prentice Hall*, 1989.

[9] B. Javidi and J. L. Horner. *Real time optical information processing.* Academic Press, 1994.

[10] K. Jensen and D. Anastassiou. Spatial resolution enhancement of images using non-linear interpolation. *Proceedings ICASSP*, 4:2045–8, 1990.

[11] A. P. Pentland. Interpolation using wavelet bases. *IEEE Trans. PAMI*, 16(4):410–14, April 1994.

[12] K. Ratakonda and N. Ahuja. Discrete multidimensional linear transforms over arbitrarily shaped supports. *To appear in ICASSP 97.*

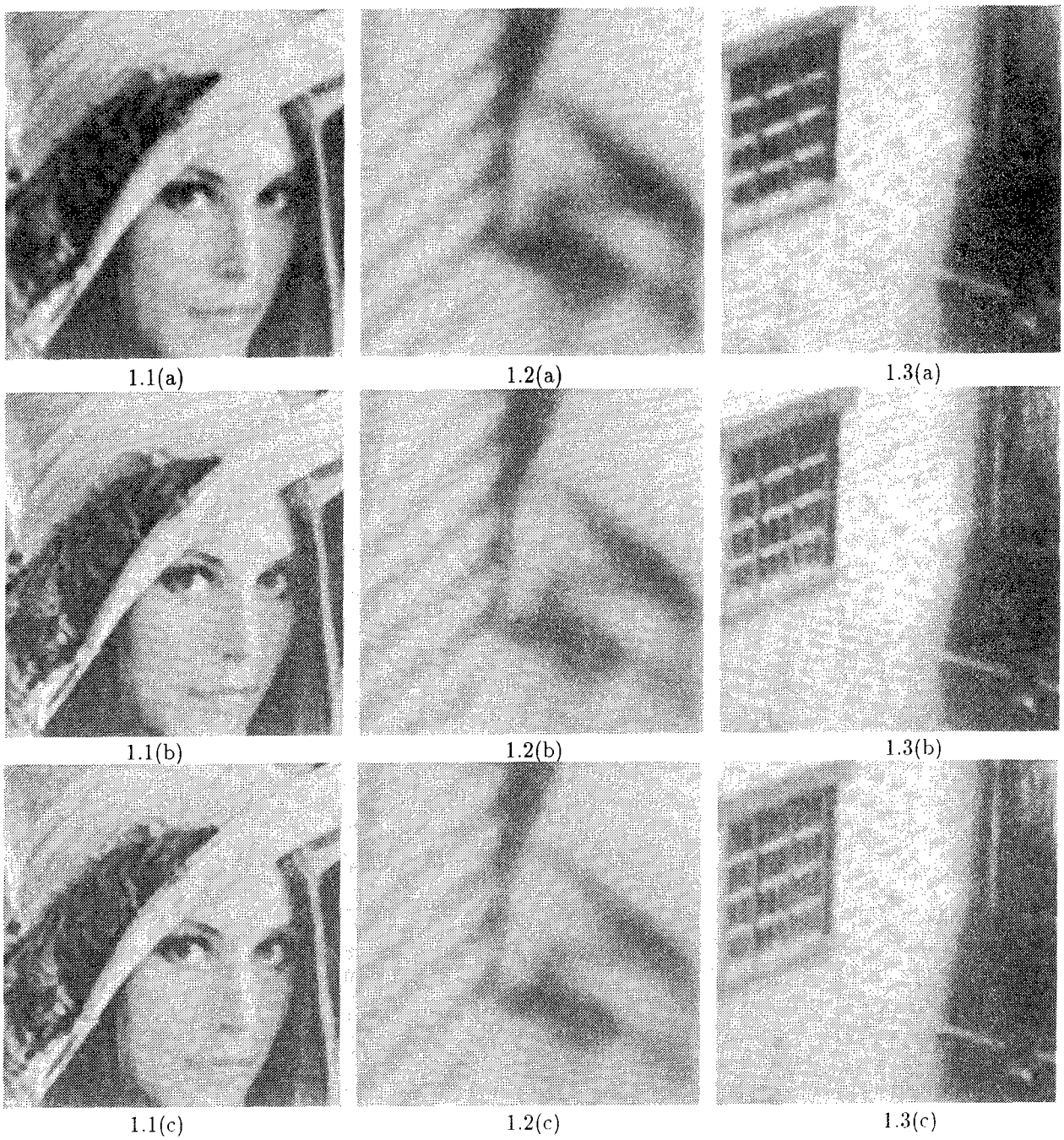


Figure 1: Resolution Enhancement results: 1.1(a) Bi-linear Interpolation 1.1(b) Proposed scheme at a coarse scale of segmentation (c) Proposed scheme at a fine scale of segmentation. Both 1.1(b) and 1.1(c) have more than 1dB improvement in PSNR over (a) when compared with original Lena image (the original small image was obtained by sub-sampling Lena by a factor of 4). 1.2-1.3 (a)-(c):Same as 1.1 (a)-(c) except that the original small image was an actual unsub-sampled clip from the Barbara (1.2 (a)-(c)) and Gold Hill (1.3 (a)-(c)).

Hierarchical Image Segmentation Using Similarity Analysis

Peter Bajcsy and Narendra Ahuja
Beckman Institute and Coordinated Science Laboratory
University of Illinois
405 North Mathews Avenue
Urbana, Illinois 61801
email: peter@stereo.ai.uiuc.edu and ahuja@vision.ai.uiuc.edu

Abstract

We present a new framework for hierarchical segmentation of multidimensional multivariate functions into homogeneous regions. Homogeneity is defined as constancy of n-th order derivatives (called features) of the function. The degree of homogeneity (scale parameter) is used to obtain multiscale segmentation. Three special types of the region hierarchies are used as tree representations of the multiscale image structure. Results showing noise robustness and computational efficiency of the proposed method are presented. Experiments to compare the method with five other segmentation techniques and applications to two- and three-dimensional images having one-, three- and six-variate data are described for the zeroth and first order region features.

1 Introduction

This paper is aimed at the fundamental problem of image segmentation. The method described extracts the hierarchical [7, 5, 1] structure present in an image characterized by varying degrees of intra-region homogeneity and inter-region contrast. The image is viewed as a multivariate multidimensional function. The goal is to partition a regular n_s-dimensional grid of sample points in the domain of a n_f-variate function into nonoverlapping connected sets of sample points forming homogeneous regions. Homogeneity is defined as constancy of n-th order derivatives (called features) of the function [6, 9]. The hierarchical segmentation problem has not been satisfyingly solved in general due to the following reasons: (1) The definition of features (e.g., the order of derivatives) is a priori unknown. (2) The amount and type of noise in the input data is unknown. Further, the computational requirements of the previously proposed methods are severe. These difficulties become more serious as the dimensionality of the data increases.

*This research was supported in part by Advanced Research Projects Agency under grant N00014-93-1-1167 administered by the Office of Naval Research and National Science Foundation under grant IRI 93-19038.

In this paper, the degree of similarity (measure of homogeneity) of two n-th order features is modeled by the Euclidean distance of their n-th order differences. It is denoted throughout the paper as homogeneity δ, which is an upper bound on the maximum difference between any pair of features within a region. The choice of order n of features depends on the application area. For perceptual purposes it suffices to use the order $n \leq 1$ [2]. The segmentation method presented in this paper uses a varying homogeneity parameter which gives rise to a number of segmentations. The noise robustness and computational efficiency of the detected regions are characterized. A tree representation of the segmentation is extracted from these segmentations that retains the subset of regions corresponding to all different scales present in the data. Experiments are reported with medical data, botanical data, satellite data, range data and gray level and color video sequences, having dimensionalities $n_s = 1, 2, 3$ (multidimensional sample space) and $n_f = 1, 2, 3, 6$ (multivariate function space), and using zeroth and first order region features.

2 Segmentation Using Homogeneity Analysis

Given the feature model, the proposed method consists of three major steps: (1) An image is segmented into regions having the same degree of feature homogeneity. (2) The degree of homogeneity is used as a scale parameter to obtain a number of n_s-dimensional (n_sD) segmentations. The detected regions split (merge) as the homogeneity parameter is made more (less) stringent and lead to a hierarchical region structure. (3) n_sD regions from the hierarchical organization are selected to obtain a tree representation of segmentations.

Mathematical framework: An n_sD image is modeled as a multidimensional multivariate function $x_i \longrightarrow f(x_i)$, where $x_i = (x_1, x_2, ..., x_{n_s})$ is a sample point and $f(x_i) = (f_0(x_i), f_1(x_i), ..., f_{n_f}(x_i))$ is the n_f-dimensional (n_fD) function value or attribute at location x_i. An n-th order feature $f^n(x_i)$ at a sample point x_i is the n-th order derivative of f (estimated by the difference), e.g., $f^0(x_i) = f(x_i)$, $f^1(x_i) =$

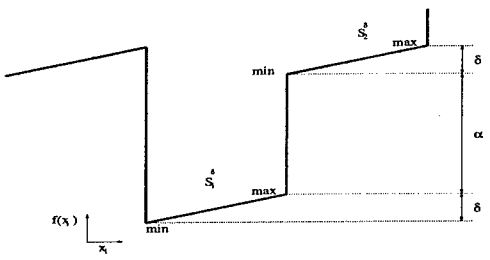


Figure 1: Homogeneity δ and contrast α for $n = 0$. The thick solid line shows a 1D cross section of $n_s D$ regions S_1^δ and S_2^δ. $\delta = \max\{ \parallel f(x_i \in S_j) - f(x_k \in S_j) \parallel \}$, $j = 1, 2$, and $\alpha = \min\{ \parallel f(x_i \in S_1) - f(x_k \in S_2) \parallel \}$.

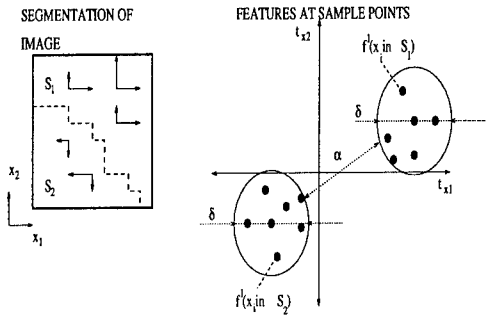


Figure 2: Homogeneity δ and contrast α for $n = 1$. The dots show derived features $f^1(x_i) = (t_{x1}, t_{x2}) = (\frac{\partial f(x_i)}{\partial x_1}, \frac{\partial f(x_i)}{\partial x_2})$ from the left image (arrows denote the sign of a derivative along that particular direction). $\delta = \max\{ \parallel f^1(x_i \in S_j) - f^1(x_k \in S_j) \parallel \}$, $j = 1, 2$, and $\alpha = \min\{ \parallel f^1(x_i \in S_1) - f^1(x_k \in S_2) \parallel \}$.

$(\frac{\partial f(x_i)}{\partial x_1}, \frac{\partial f(x_i)}{\partial x_2}, \ldots \frac{\partial f(x_i)}{\partial x_n})$. The objective of segmentation is to partition the regular n_s-dimensional grid of sample points x_i into nonoverlapping connected sets denoted as regions $S_j = \{x_i\}$, where j is the index of the region. $n_s D$ regions are defined by homogeneity (similarity) of features within the region, and contrast (dissimilarity) with the surround. The homogeneity δ of a region S_j^δ is defined as the maximum distance between a pair of features at the sample points within the region S_j^δ; $\max\{\parallel f^n(x_i \in S_j^\delta) - f^n(x_k \in S_j^\delta) \parallel\} = \delta$ (see Figures 1 and 2). The contrast α of two neighboring regions S_1^δ and S_2^δ is defined as the minimum distance between a pair of features located across the region boundary; $\alpha = \min\{ \parallel f^n(x_i \in S_1^\delta) - f^n(x_k \in S_2^\delta) \parallel \}$ (see Figures 1 and 2).

Segmentation method: (1) Create regions $S_{x_i}^{2\delta} = \{x_k\}$ at each sample point x_i such that $\parallel f^n(x_i) - f^n(x_k) \parallel \leq \delta$. (2) Compare the pair of regions $S_{x_i}^{2\delta}, S_{x_l}^{2\delta}$ for every two adjacent sample points x_i, x_l. (3) Define final regions or boundaries based on the comparisons.

Noise robustness: Mean attribute values computed over $S_{x_i}^{2\delta}, S_{x_l}^{2\delta}$ are used as region descriptors in the comparison in Step 2 to achieve noise robustness. The descriptor of $S_{x_i}^{2\delta}$ is calculated as $\bar{\mu}_{x_i} = \frac{1}{M_{x_i}} \sum_{k=1}^{M_{x_i}} f^n(x_k \in S_{x_i}^{2\delta})$. The comparison in Step 2 and assignment of samples to different regions in Step 3 are performed based on the following inequality: If $\parallel \bar{\mu}_{x_i} - \bar{\mu}_{x_l} \parallel \leq \delta$ then x_i and x_l belong to the same final region else x_i and x_l are boundary points. This grouping rule is an outcome of similarity analysis. The similarity analysis relates the values of region homogeneity and contrast statistically. Two cases are considered: $\alpha > \delta$ and $\alpha \leq \delta$. The probability of obtaining correct segmentations is obtained analytically and numerically in terms the probability of error in grouping x_i and x_l into one region. If $\alpha > \delta$ then $Pr(error) = 0$. If $\alpha \leq \delta$ then $Pr(error) = 1 - Pr(\parallel \bar{\mu}_{x_i \in S_j^\delta} - \bar{\mu}_j \parallel \leq \delta$, where $\bar{\mu}_j$ is the sample mean of features from an unknown region S_j^δ.

Computational efficiency: The dimensionality of computations is reduced using a separability property. Efficient segmentation method using the separability property is performed in three steps: (a) divide $n_s D$ grid of sample points into several lower-dimensional grids, which leads to lower-dimensional images. (b) compute segmentations of lower-dimensional images and (c) assemble computed lower-dimensional regions or boundaries into $n_s D$ regions or boundaries of regions. The use of the separability property reduces computational complexity but decreases noise robustness. Accuracy analysis of the method using descriptors and separability shows that if $\alpha > \delta$ then $Pr(error) = 0$, but if $\alpha \leq \delta$ then $Pr(error)$ is larger than for the method not using the separability.

Hierarchical segmentation: Image segmentations are obtained for different values of a homogeneity parameter δ. *A hierarchy of regions* is created such that every region S_j^δ obtained at scale δ cannot split at $\delta + \Delta\delta$ into smaller subregions (bottom-up constraint) and cannot merge at $\delta - \Delta\delta$ with other $n_s D$ regions (top-down constraint). The hierarchy is guaranteed by replacing feature values within created regions S_j^δ at each scale δ by the sample means of created regions. $g(x_i, \delta) = \frac{1}{M_j} \sum_{k=1}^{M_j} f^n(x_k \in S_j^\delta)$, where M_j is the number of samples in S_j^δ. Three types of *tree representations* are derived from the set of regions comprising the hierarchical organization. These trees consist of (a) stable regions, i.e., $S_j^\delta = S_j^{2\delta}$; (b) regions characterized by large boundary discontinuity, i.e., $\alpha > \delta$ for a part of boundary, or (c) regions participating in a merger of $S_1^{\delta_1}$ and $S_2^{\delta_2}$ at $\delta > \delta_1 + \delta_2$ but not before. Thus, a tree contains only some of the regions and links from the original hierarchy, selected to ensure a certain type of distinction among the different regions. The different levels in the tree correspond to salient structures seen in the image at different scales.

3 Performance Evaluation

Segmentation quality is judged based on three main criteria: (a) accuracy, (b) time and memory requirements and (c) comparisons with other segmentation methods. The tradeoff between segmentation accuracy and computational requirements is as follows: the proposed segmentation using descriptors is 3 times more noise robust than the proposed segmentation using descriptors and separability but 77 times slower and 7 times more memory intensive. The experiments were conducted for the zeroth and first order region features ($n = 0, 1$) with the worst case boundary error of 7.1% which occurred for the faster but less accurate method using descriptors in one-dimensional space. The average segmentation time for the faster method was $0.4s/seg$ if $n = 0$ and $1.54s/seg$ if $n = 1$ for $100x100$ image ($n_s = 2, n_f = 1$) on Sparc 20 workstation. Three other segmentation methods were compared based on the number of misclassified sample points using synthetic images with regions characterized by $\alpha = 1$ and $\delta = 49$. The best performance was achieved by the proposed method using descriptors, followed by segmentation based on Markov Random Field (MRF), the proposed method using descriptors and separability, and morphological segmentation and Canny edge detector. The comparison of two proposed methods was extended by using two synthetic images from the work of Won and Derin [8]. The results obtained by segmenting these two images (islands and berries) were compared to the results stated in [8] and [4] obtained using different implementations of segmentation based on MRF. Although the accuracy of the proposed method with descriptors and separability is worse than that of the best results given in [4] by 0.098% (islands) and 0.577% (berries), the time requirements of our method are 10 times less that in [4] and 20 times less than in [8]. The proposed methods were used in applications involving the following types of real data: angiograms (Figure 3), magnetic resonance images (Figure 4), botanical data (Figure 5), satellite images (Figure 6), range data (Figure 7) and gray-scale and color images and video sequences (Figures 8, 9 and 10).

4 Conclusions

We have presented a new framework for hierarchical image segmentation into homogeneous regions defined by constancy of the n-th order derivatives. The degree of homogeneity was used as a scale parameter to obtain a multiscale space of segmentations. Segmentation results are represented in the form of a tree of regions. We have formulated into the method a tradeoff between the segmentation accuracy and computational requirements. Comparative analysis and experiments with multidimensional multivariate real data were conducted to demonstrate the superior performance of the proposed methods.

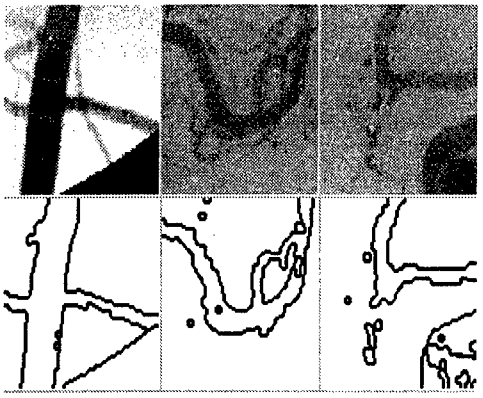


Figure 3: Results from one scale of segmentation of angiograms.
Top row - windows of size 300x300 from a large angiogram; $n_s = 2, n_f = 1$. Courtesy of the Department of Neurosurgery, College of Medicine, University of Illinois in Chicago. Bottom row - selected cross sections of a tree of detected regions ($n = 0$).

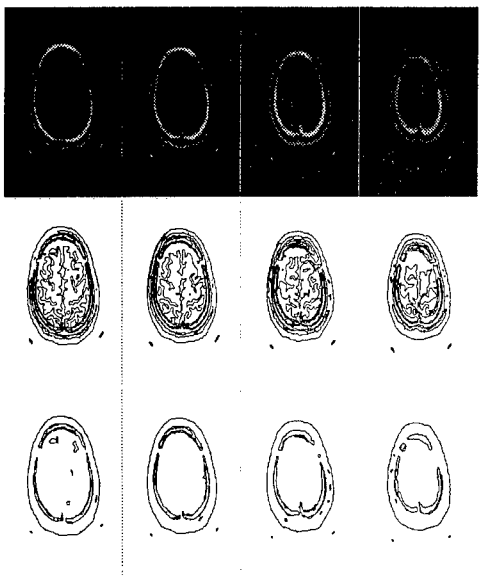


Figure 4: Results from two scales of segmentation of 3D MR data.
Top row - 2D slices of a 3D image along z axis (256x256), $n_f = 3, n_f = 1$. Courtesy of the Department of Neurosurgery, SUNY Health Sci Center, Syracuse, New York 13210. Cross sections of the 3D segmentation at (1) fine scale δ_1 (second row from top), coarse scale δ_2 (bottom row, $\delta_1 < \delta_2$) ($n = 0$).

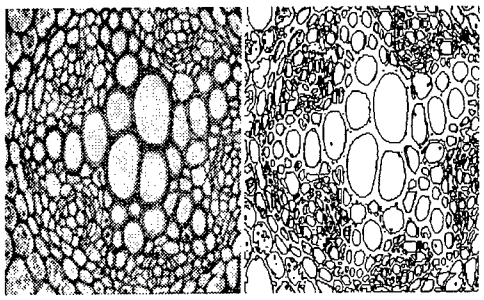


Figure 5: Results from one scale of segmentation of the 2D cross section of a plant.
Left - original image $n_s = 2, n_f = 1$. Courtesy of the Department of Plant Biology, University of Illinois at Urbana-Champaign, Illinois. Right - contours of detected regions at one scale for features with $n = 0$.

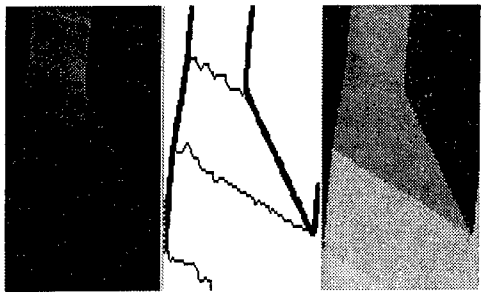


Figure 7: Results from one scale of segmentation of range data.
Left - a window of original range data $n_s = 2, n_f = 1$ obtained from [3]. Middle - segmentation result for features with $n = 1$. Right - the ground truth segmentation.

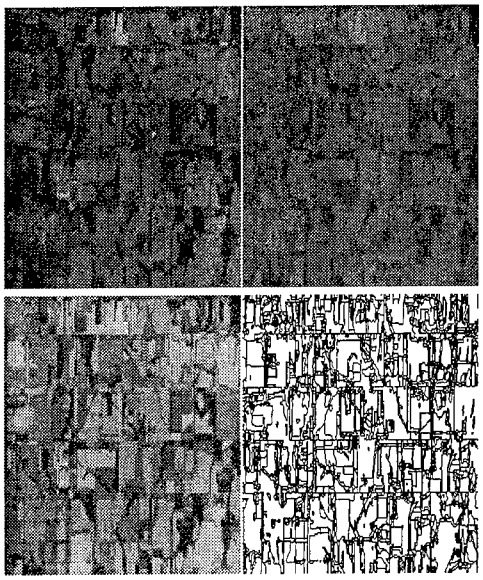


Figure 6: Results from one scale of segmentation of a satellite image.
Top - the original 6-band Landsat-5 TM collage, left - 1st, 2nd and 3rd band together, right - 4th, 5th and 7th band together, $n_s = 2, n_f = 6$. Courtesy of the Illinois Natural History Survey. Bottom - segmentation of the satellite data for features with $n = 0$. The detected regions from the tree representation are shown with their average intensity values (left) or their contours (right).

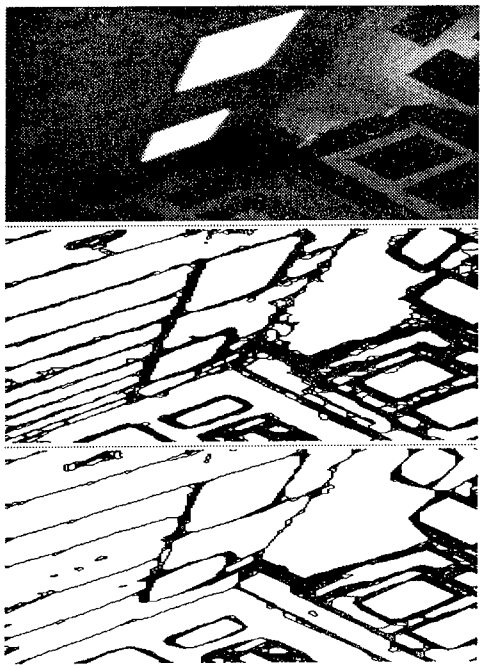


Figure 8: Results from two scales of segmentation of a gray scale image.
From top down: original $n_s = 2, n_f = 1$, segmentation at two different scales for features with $n = 1$.

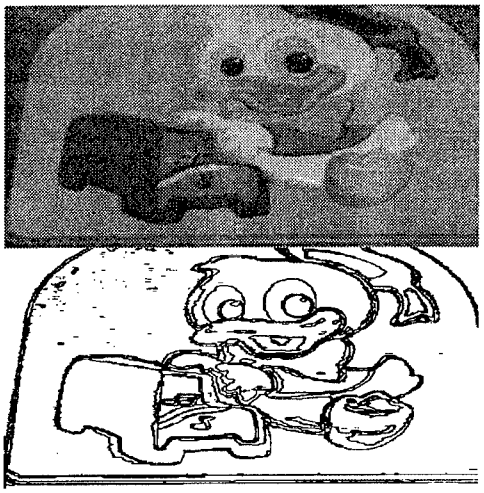


Figure 9: Results from one scale of segmentation of a color image.
Top - original $n_s = 2, n_f = 3$, bottom - segmentation result for features with $n = 0$.

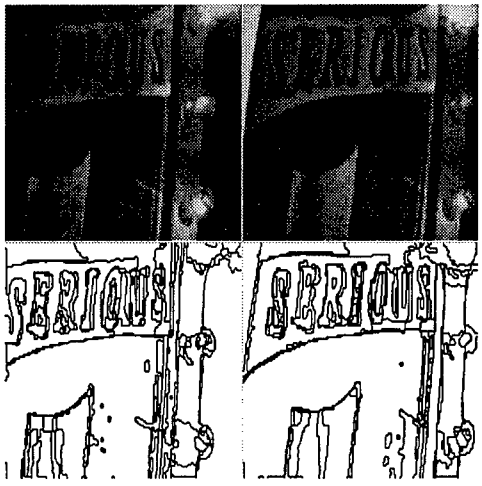


Figure 10: Results from one scale of segmentation of a color video sequence.
Top - original frames at $t_0 = 0$ (left) and $t_1 > t_0$ (right); $n_s = 3, n_f = 3$. Bottom - segmentation result for features with $n = 0$.

References

[1] N. Ahuja. A transform for detection of multiscale image structure. In *IEEE Conf. Comp. Vis. Pattern Recognition '93*, pages 780–781, New York, June 1993.

[2] T. O. Binford. Generic surface interpretation: Observability model. In *Int. Symp. on Robotics Research*, 1987.

[3] A. Hoover, G. Jean-Babtiste, et al. A comparison of range image segmentation techniques. Technical report, USF, ftp: figment.csee.usf.edu /pub/segmentation-comparison/ tech-report.95-01.ps, 1996.

[4] I. B. Kerfoot and Y. Bresler. Theoretical analysis of multichannel mrf image segmentation algorithms. *IEEE Trans. on Image processing, submitted in January 1996.*

[5] J. J. Koenderink. The structure of images. *Biol. Cybernetics*, 50:363–370, 1984.

[6] T. Pavlidis and Y. Liow. Integrating region growing and edge detection. *IEEE Trans. on Pattern Anal. and Mach. Intel.*, 12(3):225–233, March 1990.

[7] B. M. ter Haar Romeny (editor). *Geometry-driven diffusion in computer vision.* Kluwer Acamedic Publisher, Norwell, MA 02061, 1994.

[8] C. S. Won and H. Derin. Unsupervised segmentation of noisy and textured images using markov random fields. *CVGIP: Graphical models and image processing*, 54(4):308–328, 1992.

[9] S. C. Zhu, T. S. Lee, and A. L. Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multi-band image segmentation. Technical Report CICS-P-454, Center For Intelligent Control Systems, MIT, March 1995.

sarMapper: A Real-Time, Interactive SAR Tactical Mapper

John B. Hampshire II

Institute for Complex Engineered Systems (ICES)
and
Department of Electrical & Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Email: hamps@ece.cmu.edu
http://gussolomon.ius.cs.cmu.edu/hamps/IU/index.html

Abstract

sarMapper is near-real-time, interactive software for generating high-accuracy tactical ground cover maps from synthetic aperture radar (SAR) imagery using current-technology laptop computers. Such maps make it possible to automate the focus-of-attention mechanism that is the foundation of tactical image analysis, target detection, and target recognition. This document describes a proof-of-concept first instantiation of sarMapper and outlines further technical issues to be addressed in the 1997 sarMapper research effort.

1 Introduction

sarMapper is computational software and a human-machine interface that learns how to generate ground cover maps from synthetic aperture radar (SAR) images, given minimal human supervision. It is a real-time, interactive learning tool that the military (as well as scientists) can teach and subsequently use to make maps for tactical (or scientific) analysis of ground cover. sarMapper runs on current-technology laptop computers, generating detailed mega-pixel maps in one to three minutes, depending on the level of map detail specified.

1.1 Background

Current-generation military SAR platforms produce vast quantities of imagery. Ground truth for this imagery — areas in which the type and quantity of ground cover is known in detail — is both scarce and expensive to obtain. Moreover, temporal changes in ground cover mean that ground "truth" is ephemeral. Added to these facts are tactical imperatives of military operations that rely on timely, accurate maps:

- military image analysts are drowning in a sea of data for lack of a real-time ability to process the data into usable information.
- military commanders need accurate tactical maps *now*, not two hours from now.
- they need them in the field, where their troops can use them; the troops need to be able to update their maps rapidly in order to reflect the changing tactical situation in real-time.
- they must be able to do this with minimal effort and training and little or no prior information regarding the area being imaged by the SAR reconnaissance platform.

sarMapper addresses these issues by learning to

This newly-funded research is sponsored by the Defense Advanced Projects Research Agency under grant F33615-91-1-1017, monitored by the United States Air Force Wright Laboratory ATR Development Branch, Wright Patterson AFB, Dayton, OH. The views and conclusions contained in this document are the author's and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Project Agency, Wright Laboratory, the U.S. Air Force, or the U.S. government.

generate detailed ground cover maps in near-real-time with minimal human intervention, using a current-technology laptop computer. The sarMapper described herein is a proof-of-concept. The proof-of-concept was designed to show that detailed ground cover maps can be generated using a human-guided "semi-supervised" parametric learning procedure without prior knowledge of ground cover. This paper describes the learning procedure, validates the concept, and describes the underlying technical and computational framework that make its real-time, interactive implementation possible.

1.2 sarMapper Overview

The sarMapper project seeks to provide the military with two fundamental capabilities:

- Fast, automated, high-accuracy, tactical map generation.
- Tactical focus of attention.

sarMapper itself is to have the following characteristics:

- Real-time learning & map generation
- Without prior ground-truth
- On a laptop (with a CD-ROM or large external disk)
- Computational efficiency (generate a high-accuracy mega-pixel map in one to three minutes on a laptop)
- High resolution / accuracy ground cover assessment
- Interactive graphical user interface (GUI): human aids computer in initial learning phase; after learning, computer maps autonomously
- Human oversight
  - · Assess focus-of-attention warnings
  - · During sarMapper's learning phases
- Multiple
  - · Wavelength (P-, L-, C-, X-Band, etc.)
  - · Polarization (single and fully polarimetric)
  - · Spatial resolution
  - · Data Sources (e.g.,...)
    - – MSTAR public data (airborne, X-band)
    - – LL ADTS SAR (airborne, X-band)
    - – JPL AirSAR (airborne, P, L, and C-band)
    - – NASA SIR-C/X-SAR (spaceborne L, C, and X-band)
- Focus-of-Attention (FOA)
  - · Detect & locate man-made ground cover
  - · Warn human according to prior tasking
- Usable with ~1 hour training

Computational efficiency forms the core of sarMapper, allowing it to generate mega-pixel maps on a current-generation laptop in one to three minutes. Ground cover types are learned using low-complexity parametric models of RF backscatter: learning takes the form of efficient model parameter estimation, which allows ground cover types to be characterized in terms of their backscatter signature — the learning and subsequent mapping take place in near real-time, owing to the efficient, low-complexity algorithms employed. In comparison, standard maximum-likelihood map generation algorithms generate maps on a time scale of hours.

sarMapper uses *semi-supervised* learning, which obviates the need for prior ground truth. The principle behind this supervised learning procedure is straightforward: humans can discern different ground cover types in a SAR image by the differences in their appearance in the image. Different ground cover types in a single-polarization image will appear to have different shades and/or textures of gray; in a false-color composite of multiple polarizations, different ground cover types will appear in different colors. Consequently, a human can identify regions of different ground cover in an image and label these regions without knowing what the different ground cover types are — using pseudonyms for the unknown ground cover classes. These pseudo-classes of ground cover can be learned, and their backscatter signatures can then be used to generate a high-resolution pseudo-map over a wide-area in the vicinity of the image used for learning. Many of the unknown ground cover types can be inferred by an image analyst from context, site-invariant backscatter signatures, historical imagery, or focussed follow-on surveys conducted using the pseudo-map to target specific survey sites. The critical characteristic of semi-supervised learning is that it can generate a useful map in real-time without prior knowledge of the area; missing details can be filled in as they are obtained, without having to re-learn or re-map the area.

Since man-made ground cover tends to backscatter little RF energy (as in the case of

obliquely illuminated metal or concrete surfaces) or substantial RF energy (as in the case of trihedral reflectors common to military vehicles), semi-supervised learning is compatible with sarMapper's focus-of-attention (FOA) mission. Very large areas (hundreds to thousands of square kilometers) can be mapped and surveyed for small areas of potential tactical interest by a single human and sarMapper team. Survey time for a thousand square kilometer area (10-meter map resolution) requires between ten minutes and one-hour on a single laptop computer, depending on the number of ground cover classes enumerated; survey time for a fifty square kilometer area requires between 15 seconds and three minutes under the same conditions.

1.3 Proof-of-Concept

The proof-of-concept objectives were to

- generate detailed ground cover maps from JPL AirSAR images *without* prior knowledge of ground cover.
- validate Tobit parametric models of radio frequency (RF) backscatter envelope at P, L, and C bands.
- show that these Tobit models can be learned with human supervision, even though the human has no prior knowledge of the ground cover at the site being mapped.
- show that Rayleigh-compressed SAR images (see section 2.2) and their associated Tobit-generated maps (section 2.5) can be used to target a small, well-defined, set of locales in a site (or set of sites) where ground truth should be surveyed.
- show that the Tobit-generated map can be converted to a true ground cover map with a few key strokes, once ground truth has been established at the selected survey sites.
- validate a number of image processing and computational advantages derived from the use of Tobit parametric models for ground cover.
- point the way to a highly automated focus-of-attention version of sarMapper.

2 Technical Summary

Figure 1 illustrates the sarMapper graphical user interface (GUI). In this paper, sarMapper takes SAR images that have been decompressed from the JPL AirSAR compression format described in [DuBois-87,AirSAR-90], but sarMapper is designed to work with arbitrary SAR platforms. The main control panel has controls for specifying the receive and transmit polarizations to be synthesized when the complex floating-point SAR data is compressed into a real 8-bit integer image. Common polarizations (e.g., HH, HV, VV, Left Circular, and Right Circular) are synthesized (when fully polarimetric data are available) by efficient code, tailored to the specific polarization. Arbitrary polarizations are synthesized via a general compression-and-synthesis algorithm, which — in the case of JPL AirSAR — is a modified ANSI-C version of FORTRAN code supplied by Eric Rignot and Pascale DuBois of JPL, section 334. Tools for labeling SAR images are above the polarization controls; these are used in the semi-supervised learning procedure described in section 2.3.

A browser (upper right corner of figure 1) shows all the SAR data files currently loaded; when a file name is highlighted, the contents of its header are displayed in an information window, which allows string searches (so the user can quickly find information of interest). Multiple SAR images can be generated from the compressed SAR data files and displayed simultaneously. Figure 1 contains two false color images derived from C- and P-band AirSAR data over NASA's Raco, Michigan super site. Each of these images overlays HH (red), HV (green), and VV (blue) polarized images to form the color composite (see the author's IU web site for a color version of this figure). The lower-right window contains locale-specific histograms for each of these three polarizations in the P-band image: they and their corresponding parametric models, listed in the lower portion of the window, are related to the learning and map generating functions of sarMapper described in sections 2.1, 2.4, and 2.5.

2.1 Parametric RF Backscatter Models

sarMapper uses SAR amplitude (i.e., RF envelope) images to generate ground cover maps. Envelope statistics are used to derive parametric models of radio frequency (RF) backscatter; when the SAR is fully-polarimetric, three parametric models are derived for each ground cover class specified for a site: one model for each of the three polarizations (HH, HV, and VV) that comprise a color-composite image. One parametric model is generated when the SAR is

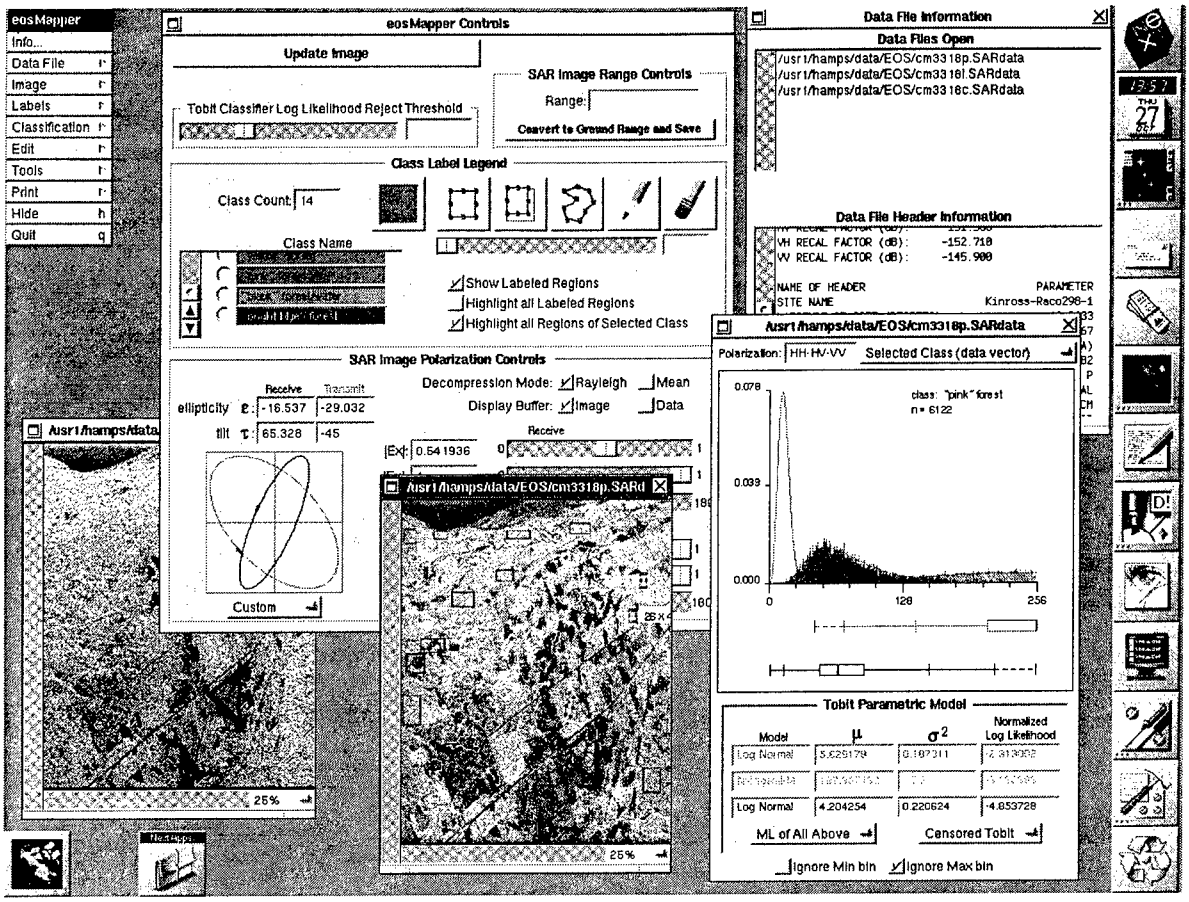


Figure 1: The sarMapper graphical user interface (GUI). The main controls allow the human operator to specify the polarization of the SAR image when the SAR is fully polarimetric, the number and type of ground cover classes, image labeling modes, etc. Multiple images can be loaded and displayed simultaneously (HH-HV-VV composites are shown for C- and P-band images of NASA's Raco Michigan super site imaged by the JPL AirSAR platform). Histograms of the backscattered SAR radio frequency (RF) envelope can be generated from the entire image, a single region, or all the regions belonging to a user-specified ground cover class: These histograms are used to generate Tobit maximum-likelihood parametric models for each ground cover class, which are in turn used to generate a ground cover map for the site and surrounding sites.

single-polarization. sarMapper models backscatter envelope (a real-valued scalar random variable); it does not model backscatter as a complex (in-phase and quadrature) random variable, as is common in the SAR literature (e.g., [Kong-87]). The reason for this is complexity. Parametric models are derived from one-dimensional histograms of RF backscatter when the envelope is used[1]; when complex backscatter is used, the histograms are two dimensional. Assuming eight-bit precision, a (one-dimensional) envelope histogram has 256 bins, whereas a (two-dimensional) complex backscatter histogram has 256^2 = 65,536 bins. Consequently, the training sample size necessary to generate a relatively smooth complex histogram — from which a low-bias, low-variance parametric model can be inferred — is, roughly speaking, the square of the size necessary to generate a relatively smooth envelope histogram and associated parametric model. Moreover, the computational complexities of evaluating log-likelihood functions — a necessary step in estimating parametric models — are proportional to the dimensionality of the histogram used: in short, using RF envelope, sarMapper avoids Bellman's curse of

[1]See section 2.6 for the rationale behind the use of histogrammed versus full-precision amplitude statistics in the parametric modeling procedure.

dimensionality [Bellman-61,Duda-73]. The computational and sample complexities of the parametric modeling procedure are orders of magnitude lower than they are for the complex backscatter paradigm (see section 2.6), so the resulting parametric classifier is significantly more efficient in its data and computational requirements.[1]

sarMapper uses Rayleigh, Rician (aka Rice-Nakagami), Nakagami-M, and LogNormal parametric models of the backscattered RF envelope. The Rayleigh, Rician, and Nakagami-M models are progressively general ones governing the statistics of the backscattered RF envelope when the complex backscatter is Normally distributed: a concise and well-formulated derivation of these models is given in [Beckmann-67]. The LogNormal parametric model spans a wide range of disciplines from econometrics to RF propagation; see [Strohbehn-75] for a motivation and description relevant to SAR backscatter. The probability density functions (pdfs) of these models follow, along with references: exhaustive details of all of them can be found in the appendices of [Hampshire-88]. In all the following pdfs α denotes the backscatter envelope random variable (rv), and A denotes the domain of α (i.e., $\alpha \in A$); $E_A[\alpha]$ denotes the expected value of α over its domain; and $f_\alpha(a)$ denotes the pdf of the rv α evaluated at $\alpha = a$.

Rayleigh pdf: α is distributed according to [Strutt-29,Davenport-58,Beckmann-67,Ishimaru-78]

$$f_\alpha(a) = \frac{a}{\sigma^2} \exp\left(-\frac{a^2}{2\sigma^2}\right), \qquad (1)$$

where $2\sigma^2 = E_A[\alpha^2]$. Note that the Rayleigh pdf's mean is given by

$$E_A[\alpha] = \sqrt{\sigma^2 \cdot \frac{\pi}{2}} \qquad (2)$$

[1]Here I use the term "efficient" in both the classical Cramér-Rao context [Rao-45,Cramér-46] and the related pattern recognition context I have defined elsewhere [Hampshire-93a,Hampshire-93b].

Rician pdf: α is distributed according to [Rice-44,Nakagami-60,Beckmann-67]

$$(3) \qquad f_\alpha(a) = \frac{a}{\sigma^2} \exp\left(-\frac{a^2 + a_0^2}{2\sigma^2}\right) I_0\left[\frac{a \cdot a_0}{\sigma^2}\right],$$

where a_0 is the value of the deterministic component of α, σ^2 is a variance-like parameter, and $I_0[\cdot]$ denotes the modified Bessel function of order zero [Abramowitz-70].

Nakagami-M pdf: α is distributed according to [Nakagami-60,Beckmann-67]

$$(4) \qquad f_\alpha(a) = \frac{2m^m \cdot a^{(2m-1)}}{\Gamma(m) \cdot \Omega^m} \exp\left(-\frac{ma^2}{\Omega}\right),$$

where $\Omega = E_A[\alpha^2]$, m is an inverse normalized variance parameter, and $\Gamma[\cdot]$ denotes the gamma function [Abramowitz-70].

LogNormal pdf: α is distributed according to [Aitcheson-66,Strohbehn-75]

$$(5) \qquad f_\alpha(a) = \frac{1}{\sqrt{2\pi}\, a\, \sigma_{\ln(\alpha)}} \exp\left[\frac{(\ln(a) - \gamma)^2}{2\sigma^2_{\ln(\alpha)}}\right]$$

where $\gamma = E[\ln(\alpha)]$ (i.e., γ is the mean log-envelope) and $\sigma^2_{\ln(\alpha)} = E_A[\ln(\alpha)^2] - \gamma^2$ (i.e., $\sigma^2_{\ln(\alpha)}$ is the variance of the log-envelope).

2.2 Rayleigh Image Compression & Equalization

SAR signals have dynamic range on the order of 55 dB[2], but an eight bit number encodes at most about 53 dB of dynamic range. Consequently, when a SAR image is displayed on a typical computer monitor, many of the pixels are

[2]Personal communication regarding the dynamic range of JPL AirSAR from J. Van Zyl, November, 1994.

saturated, taking on the minimal value of 0 or the maximal value of 255.

RF envelope (amplitude) displays of SAR imagery are formed by converting the complex backscatter amplitude (in-phase and quadrature components are generally represented with 32-bit floating-point representation or some derivative of this format) into a real magnitude represented with an 8-bit integer. If this compression is done with too much gain the resulting image is very bright; if it is done with too little gain, the resulting image is very dark. In either case, image detail is lost (either due to high-end or low-end saturation). The key to good image contrast and brightness is a compression scheme that minimizes the amount of both high- and low-end saturation while preserving as much of the dynamic range contained in the complex floating-point representation. From an information-theoretic perspective, the compression scheme that minimizes saturation and maximizes dynamic range minimizes the information loss between the complex floating-point and real integer image representations.

Typically, SAR images are compressed so that the mean pixel value is some target value in the vicinity of 128 (half of 8-bit full scale). The mean pixel value is estimated by sub-sampling the image with some default gain factor. Then the full image is compressed using a new gain factor that results in a mean pixel value close to the specified target value. This works well if the pixels are Normally distributed, but they are not. Rather the pixels are generally distributed according to one of the pdfs described in the previous section, all of which are characterized by a fairly high level of skewness. The histogram in the lower left image of figure 2 illustrates. It is based on the HH-polarized image generated from a JPL AirSAR L-band image of the Flevoland super site in The Netherlands (top, left). The image was generated using the standard mean-based compression scheme just described. The whole image histogram is approximately LogNormally distributed, with a very long tail: three quarters of the pixels have values less than 120, but more than ten percent of all pixels have the maximum value of 255. This is because the mean-based compression algorithm tends to set the compression gain too high in its attempt to match the mean pixel value with the target value (again, owing to the skewness of the RF envelope pdfs described above).

Rayleigh compression works somewhat differently. Instead of trying to match the mean pixel value with some target near half scale, it determines the compression gain factor so that the fraction of pixels with the maximum value of 255 approximates a pre-determined target value of 5%. I call this value the "target saturation fraction". Matching the actual fraction of saturated pixels to the target fraction ensures that the image is neither too bright nor too dark. The compression scale factor needed to realize the target saturation fraction is computed by sub-sampling the image, fitting a Rayleigh pdf to the resulting histogram, and estimating the *cumulative* distribution function (cdf) for the data (under the Rayleigh assumption). The cdf can then be used to determine the compression scaling factor necessary to achieve the target saturation fraction. Here is how it works...

We have the sub-sampled image pixels, which we view as realizations of the RF envelope backscatter rv α. We compute the maximum-likelihood estimate of the Rayleigh pdf's parameter σ^2 in (1). Let's call this estimate $\sim\sigma^2$. Next we compute the Rayleigh cdf for the high-end saturation amplitude minus one ($\alpha = 254$) — which we denote by $\zeta_\alpha(254)$ — given $\sim\sigma^2$. This is given by

$$\zeta_\alpha(254) = 1 - \exp\left(-\frac{(255)^2}{2\sim\sigma^2}\right) . \tag{6}$$

We can then compute the compression scaling factor δ that we should use so that the cdf of the re-scaled backscatter rv $(\delta\cdot\alpha)$, evaluated at the saturation amplitude minus one (254), is equal to one minus the target saturation fraction f_s:

$$\zeta_{\delta\cdot\alpha}(254) = 1 - \exp\left(-\frac{(255)^2}{2\sim\sigma_\delta{}^2}\right) \cong 1 - f_s \tag{7}$$

Solving (7) for the Rayleigh parameter $\sim\sigma_\delta{}^2$, which corresponds to the target saturation fraction f_s, we obtain

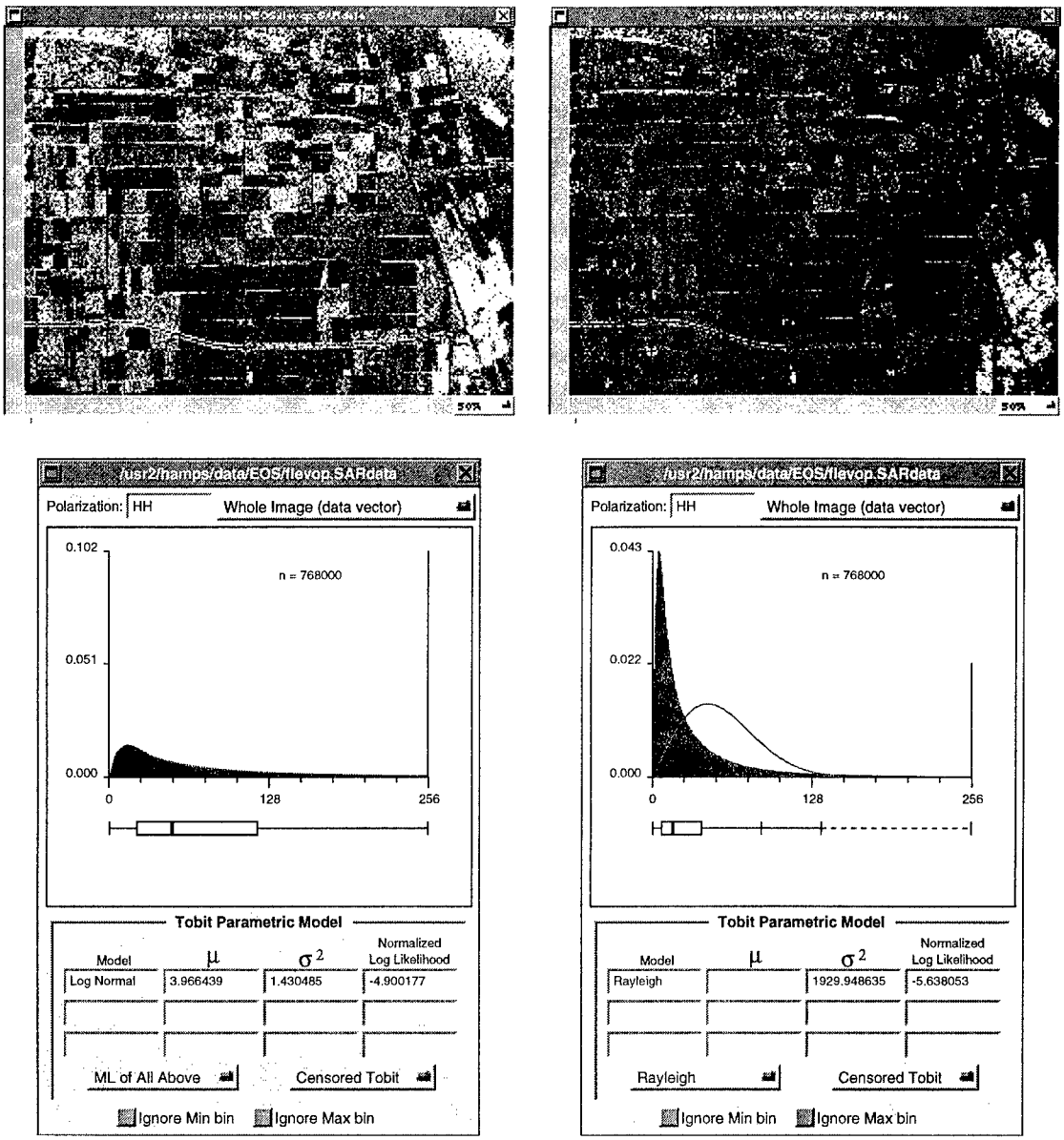


Figure 2: Two HH-polarized images generated from a JPL AirSAR L-band image of the Flevoland super site in The Netherlands. The left-hand image was generated using mean-based compression: the whole-image histogram below the image indicates that about ten percent of the image pixels are saturated (i.e., they have the maximum value of 255). The large number of light regions in the image confirms this. The right-hand image was generated using Rayleigh-based compression in which the image saturation was limited to Order[5%]: the whole-image histogram below indicates that only about two percent of the image pixels are saturated. The resulting monochromatic HH image is darker and shows less detail that its mean-compressed counterpart in this grayscale version, but the associated HH-HV-VV color composite image has better contrast and color balance. More importantly, the statistically consistent balance between image contrast and brightness afforded by Rayleigh compression equates to consistent dynamic range in the 8-bit data vector used for parametric learning and classification. This, in turn, leads to more consistent parametric models for ground cover classes and more accurate maps.

$$\sim\sigma_\delta^2 = -\frac{(255)^2}{2\ln(f_s)} \quad . \tag{8}$$

Given (2), we find that the scaling factor that will generate f_s is given by

$$\delta = \frac{E_A[\delta \cdot \alpha]}{E_A[\alpha]} = \frac{\sim\sigma_\delta}{\sim\sigma} = \sqrt{\frac{-(255)^2}{2\ln(f_s) \cdot \sim\sigma^2}}; \; 0 < f_s < 1 \tag{9}$$

The right-hand side of figure 2 shows the effect of Rayleigh compression. The scaling factor δ is computed for a target saturation fraction $f_s = .05$ and the image is compressed. After compression, the whole-image histogram is generated (bottom, right) and we find that the actual saturation fraction is about 2%. The difference between the target value of 5% and the actual value of 2% follows from the actual distribution of the whole-image data. In computing the scaling factor, we *assumed* that the whole image is Rayleigh distributed, when *in fact* it is approximately LogNormally distributed. Because the Rayleigh pdf isn't a proper model of the data, our computations are biased, and our actual saturation fraction is smaller than the 5% target.

This requires an explanation of our rationale for using only the Rayleigh pdf to compute the scaling factor instead of considering other models: *the Rayleigh cdf can be computed in closed form.* Consequently, the scaling factor δ can also be computed in closed form (i.e., very efficiently). What we lose in an imprecise estimate of the scaling factor necessary to achieve the target saturation fraction, we make up for in speed (the whole scaling factor estimation procedure for a mega-pixel SAR image takes a fraction of a second). The Rayleigh pdf is generally sufficient to get us within plus or minus 3% of our target saturation fraction of 5%.

Rayleigh compression generates images with *statistically consistent* dynamic range and saturation; mean-based compression generates images with dynamic range and saturation that can vary significantly with the skewness of the whole-image histogram. This statistical consistency contributes to statistically better ground cover maps generated from the Rayleigh compressed images.

One final note on the Rayleigh compression algorithm... The HH-polarized image is sampled to compute the scaling factor because it is consistently the one with the highest dynamic range. It is sub-sampled by sampling *all* the pixels in every 17th line in the image, beginning with the 23rd image line — these numbers are intentionally prime numbers, so the chance of computing a bad estimate of the whole-image statistics due to some periodic artifact is reduced.

2.3 Semi-Supervised Learning

sarMapper was conceived with two basic assumptions:

- There is usually little or no ground truth associated with a SAR site to be mapped.
- SAR backscatter statistics correlate strongly with the physical structure of the illuminated ground cover, and parametric models of the backscattered RF envelope are well understood (section 2.1).

The lack of ground truth implies an un-supervised learning procedure, but the strong prior knowledge we have about the statistics of radar backscatter suggests a parametric modeling paradigm, which involves supervised learning. I concluded that it would be possible to design a "semi-supervised" learning procedure that would use parametric models without prior knowledge of ground cover. What I will describe in this section is the first instantiation of this semi-supervised learning, which involves a human supervisor. The critical reader will note that human-supervised learning is a risky business because the supervision is subjective (we are non-deterministic creatures) and it varies from human to human (we are all different). Statistically, therefore, the level of human involvement in this first instantiation of semi-supervised learning is a weakness, but it conveys two advantages:

- It exploits the strong inferential capabilities of the human supervisor.
- It is fast and highly interactive. The user can interact with the machine and see the results of this interaction in real-time. For example, he can make a change to the ground cover regions specified for the machine's learning phase, have the machine re-learn the regions,

and generate a new map — all in about two minutes.

How does the human user identify these homogenous regions? sarMapper generates a color composite image of the site by overlaying an HH image in red, an HV image in green, and a VV image in blue in the case of fully-polarimetric radar (or a single grayscale image of the HH polarization in the case of single-polarization radar); igure 3 shows such an image, derived from a JPL AirSAR P-band image of the Landes maritime pine forest in France. The user then identifies a number of regions in the image with the same basic color (or shade of gray), outlines them or colors them in, and groups them into one so-called ''pseudo-class''. This process is repeated for each pseudo-class identified in the composite image by the human supervisor: each pseudo-class is assigned a map color and a pseudonym. sarMapper's main control panel, shown in figure 1, contains the region labeling controls. Regions representing a given ground cover pseudo-class can be outlined using one of three modes (square region, rectangular region, or polygon) or they can be ''penciled in'' — a fourth labeling mode that identifies small, oddly-shaped regions (e.g., winding rivers) by coloring in a local binary bitmap. The number of ground cover classes, their names, and their map colors are user-specified to the right of the labeling controls in figure 1.

Once the training regions for each pseudo-class have been identified by the human supervisor, sarMapper characterizes each ground cover pseudo-class by computing the maximum-likelihood parameters for all four possible backscatter parametric models (Rayleigh, Rician, Nakagami-M, and LogNormal) — it does this for each of the three polarizations (HH, HV, and VV). The parametric model with the greatest log-likelihood (summed across the three polarizations) is then chosen as the maximum-likelihood (ML) model for the ground cover pseudo-class. Summing the log-likelihoods implicitly assumes that the HH, HV, and VV polarized images are independent, which they are not; the procedure is taking a statistical liberty for the sake of computational simplicity. The resulting ML model characterizes its associated ground cover pseudo-class and is used — along with all the other ML models — to produce a map (section 2.5). Figure 3 shows how the color composite image looks when a number of pseudo-classes (note the names) have been identified and outlined. At this point, sarMapper is ready to generate Tobit ML parameter estimates for all the candidate pdfs, in order to determine the ML model for each pseudo-class.

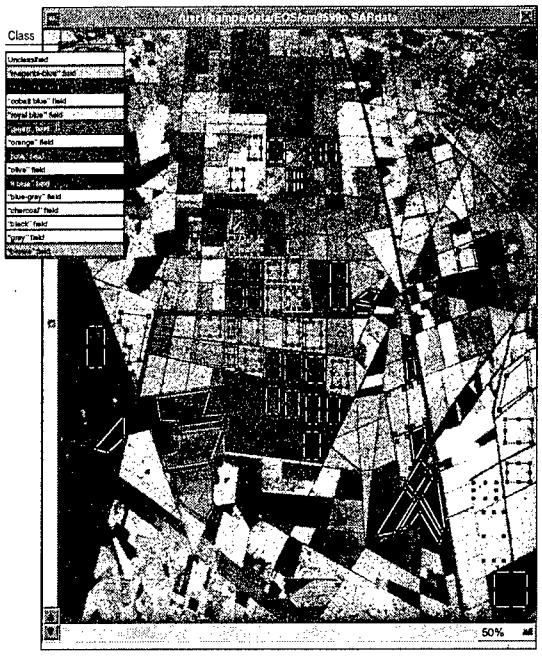


Figure 3: A sarMapper-generated HH-HV-VV false-color composite image of the Landes forest in France. The composite (shown here in grayscale — see the author's IU website for a color version) was generated from a JPL AirSAR P-band image. A human labels distinctly colored regions in the case of a fully-polarimetric radar (or shaded ones in the case of a single-polarization radar) denoting distinct backscattering properties in the image. These regions are shown above with ''hooks'' used to move/resize them. The regions are given color codings and pseudonyms (since the actual type of ground cover is generally *not* known *a priori*). The legend in the upper left corner of the figure shows these pseudonyms and color codings. sarMapper then learns a Tobit maximum-likelihood parametric model for each ground cover class and generates a color-coded ground cover map of the entire site and its surrounding areas.

In the next two sections I will describe how Tobit parametric estimation works, how Tobit models

are used to generate ground cover pseudo-class maps, and how these pseudo-maps are converted into true ground cover maps.

2.4 Tobit Estimation of Ground Cover Class-Conditional Densities with RF Envelope Backscatter Models

In section 2.1 I described the parametric models that sarMapper uses to characterize the backscattered RF envelope and I hinted why sarMapper uses eight-bit data representations and real backscatter (amplitude) histograms, rather than 32-bit floating point data representations of complex backscatter (I explain this fully in section 2.6). In section 2.2 I explained that Rayleigh compression is used in order to control the amount of saturation in the SAR images synthesized during compression of the JPL AirSAR data. Because the backscatter contains at least 2 dB more dynamic range than can be encoded with an eight bit number, there is *always* some image saturation, which raises the following question: how does sarMapper develop good parametric models of regions that generate heavily saturated RF backscatter statistics? The answer: by using a Tobit (i.e., Tobin probit) maximum-likelihood parameter estimation procedure.

A detailed explanation of Tobit estimation is well beyond the scope of this paper, so I will discuss it in very general terms. Tobit estimators were first developed in the 1940's by Hald [Hald-49], but the work went un-recognized. When it was developed independently and published almost a decade later by Tobin [Tobin-58], it was called the ''Tobin-probit'' model, a name that was later shortened to ''Tobit model''. Amemiya has written extensively on Tobit models in the econometrics context; his most general description can be found in [ch. 10, Amemiya-85]. I describe Tobit models in the RF backscatter context in [Hampshire-88,Hampshire-92]. Given a random variable α with a known pdf, the corresponding Tobit model is the pdf for the random variable α', which α becomes when it is measured by a saturating device. In its simplest form, a high-end saturating device produces α' by ''clipping'' α for all values above the saturation threshold α_τ:

$$\alpha' = \begin{cases} \alpha, & \alpha < \alpha_\tau \\ \alpha_\tau, & \text{otherwise} \end{cases} \qquad (10)$$

Audiophiles will recognize the clipped output of an over-driven amplifier as a saturated random variable. The audio mathematics are more complicated because the transition from linear to non-linear amplifier operation is gradual, *not* discontinuous, as it is in (10).

When computer images are generated from SAR data, the backscattered RF envelope α is converted to a saturated, eight-bit-quantized envelope α'. Equation (10) is a reasonable description of the saturated RF envelope prior to eight-bit quantization. Rayleigh compression limits the amount of saturation in the whole image, but it does not limit the amount of saturation in localized regions of the image. As a result, some regions within an entire SAR scene are heavily saturated. Figure 4 shows the histogram for a heavily saturated region in a JPL AirSAR P-band image of the Raco, Michigan super site (HH-polarization). The left view shows the full histogram, and the right view shows the un-saturated part of the histogram. Thirty four percent of the region's pixels are saturated, so the histogram is dominated by the statistics of its most significant bin ($Q(\alpha') = 255$, where $Q(\cdot)$ denotes the 8-bit quantization operator). This is clear in the left-hand view. If we ignore the most significant bin, we see that the un-saturated part of the histogram still has a recognizable shape. When a LogNormal Tobit estimator is applied to the full, saturated histogram, the resulting parametric model fits the data well (a plot of the model is superimposed on the histogram in the right-hand view). Mathematically, the Tobit estimator combines what can be inferred from both the un-saturated and saturated parts of the histogram to estimate the parametric model of the *un*-saturated rv α from its quantized, saturated counterpart $Q(\alpha')$. In fact, Tobit estimators for the parametric models described in section 2.2 are *efficient* maximum-likelihood estimators (in the Cramér-Rao sense [Cramér-46,Rao-45]) [Hampshire-92]. They yield good models even when the amount of saturation is substantial (as it is in figure 4). Since sarMapper learns Tobit parametric models of the 8-bit quantized RF backscatter envelope to characterize each ground cover class, it is able to

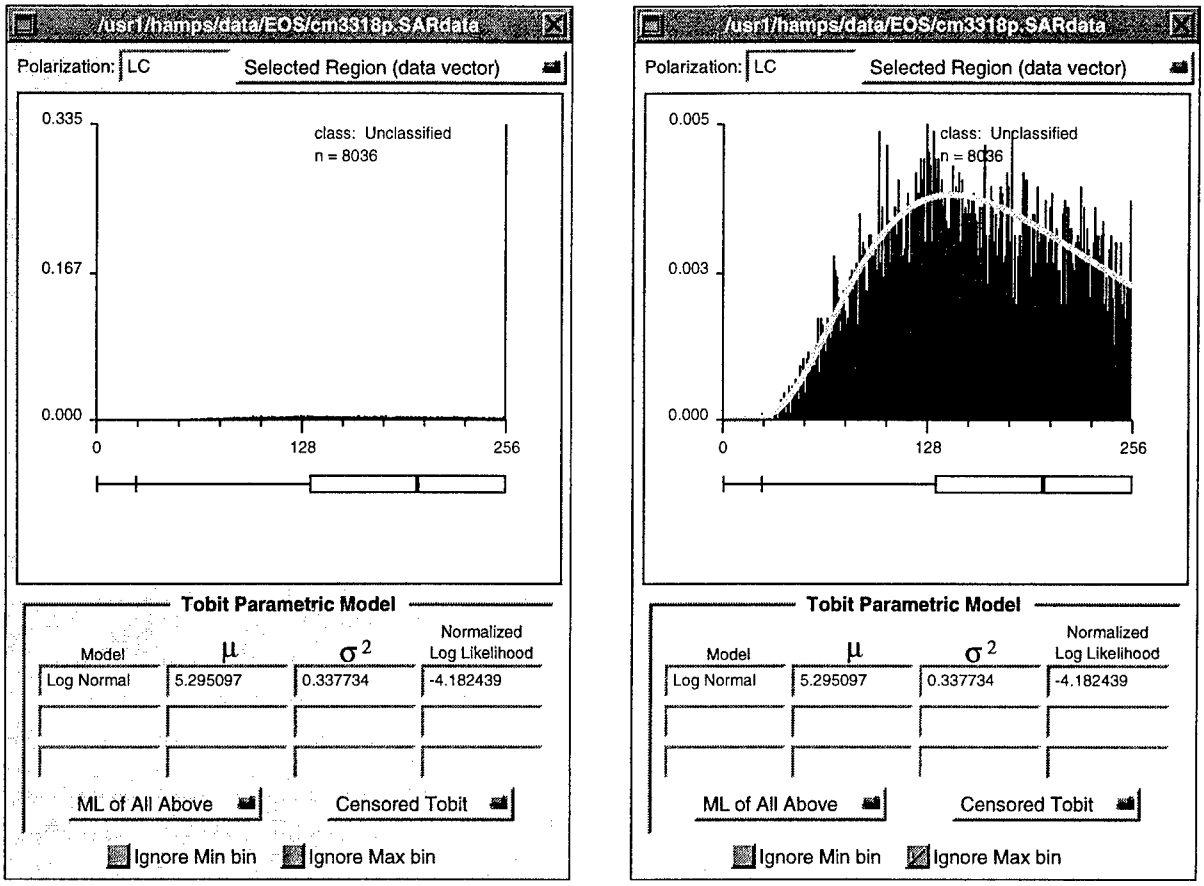


Figure 4: Two views of one histogram, which characterizes the RF envolope (JPL AirSAR, P-band, HH-polarized) backscattered from a region in NASA's Raco, Michigan super site. The left-hand view shows the full histogram: about 34% of the HH image pixels in the region are saturated. The right hand view shows the un-saturated part of the histogram (i.e, the histogram for bins 0 – 254). About 66% of the pixels fall in this range. The Tobit ML estimation procedure found that the histogram was best fit by the LogNormal pdf using the parameters listed below the display. Note that the model (superimposed on the histogram) fits the histogram well, despite the heavy amount of saturation. This illustrates two important facts: an 8-bit histogram of the RF backscatter envelop remains a sufficient statistic for inducing the parametric model of the backscatter despite substantial saturation; the Tobit estimator remains efficient (in the Cramér-Rao sense) under substantial saturation. A standard maximum-likelihood parameter estimation procedure for the LogNormal pdf would yield a very poor fit to this saturated data (see [Hampshire-92]).

---

generate good maps from eight-bit amplitude images in near real-time.

The histograms and box plots [Tukey-77] for each ground cover pseudo-class are generated by compiling statistics on the values of all the pixels belonging to that class: this is done for the three (fully-polarimetric radar) or single (single-polarization radar) 8-bit (i.e., 256-value) plane(s) in the sarMapper data vector. For fully-polarimetric radar these correspond to the Rayleigh-compressed HH (red), HV (green), and VV (blue) polarized data vectors for the image; for single-polarization the Rayleigh-compressed HH image is displayed in a single grayscale image plane. The ML Tobit model is then computed for each of the pseudo-classes as described in the previous section. The Tobit estimation procedures described in [Hampshire-88,Hampshire-92] govern the computations by which each model's parameters and log-likelihood are estimated.

The training data and computational

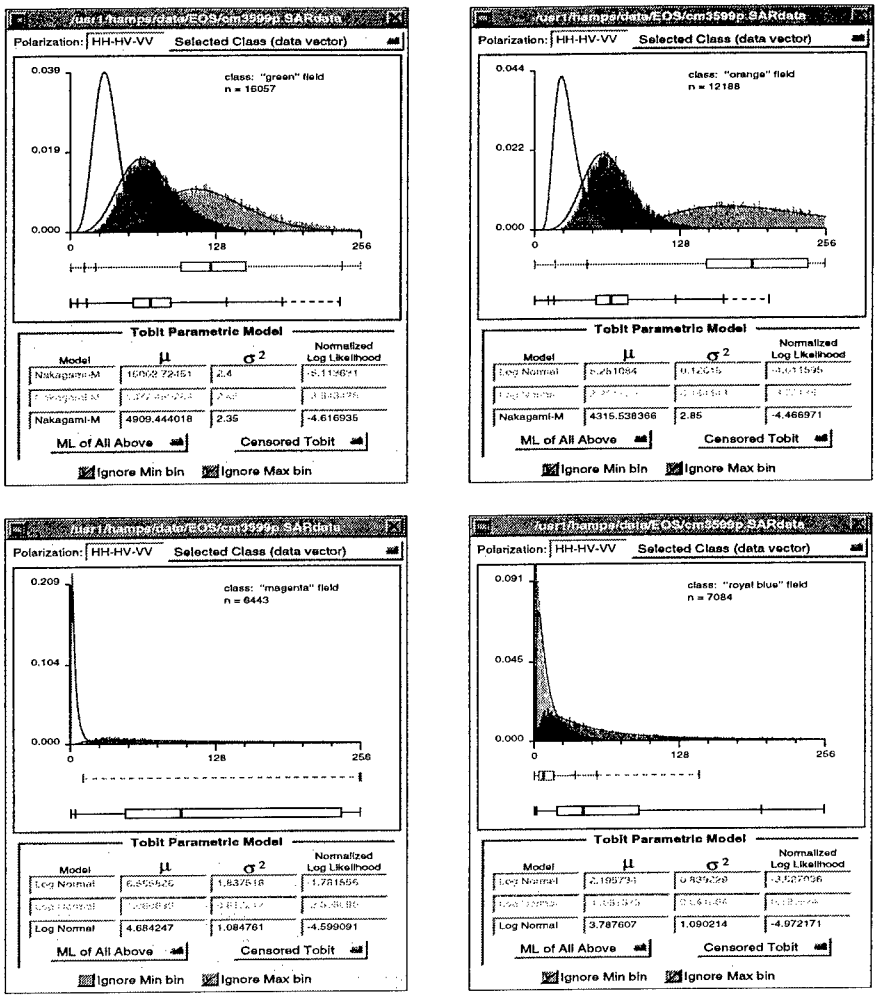


Figure 5: Histograms, box plots [Tukey-77], and the resulting Tobit maximum-likelihood (ML) parametric models for four ground cover pseudo-classes in the Landes maritime pine forest P-band image (see figure 3). The box plots and list of Tobit ML models are arranged as follows for each pseudo-class: HH (top), HV (middle), VV (bottom). The histogram polarizations are not obvious in a grayscale version of this figure, but can be inferred from their corresponding box plots. In general the HH histogram is the right-most (i.e., the one with the largest mean/median/mode), the VV is the central one, and the HV is the left-most (i.e., the one with the smallest mean/median/mode) for each pseudo-class. The Tobit ML parametric model for each polarization is superimposed as a solid line on its associated histogram. Nakagami-M and LogNormal models are the common (although not exclusive) ML choice since they represent more general RF scattering paradigms. Note that the models typically fit the histograms well. The Tobit parametric estimators are robust, despite the saturation that inevitably occurs when a radar signal is modeled with only 8-bit precision.

requirements of sarMapper *are reduced by about four orders of magnitude* when the low-complexity eight-bit/real data representation I have described is used instead of the 32-bit/complex representation commonly used. Tobit models allow sarMapper to exploit the efficiencies afforded by the low-complexity data representation without significant reductions in map accuracy or precision. Figures 5 illustrates why this is so: it shows the maximum-likelihood Tobit parametric models for four ground cover pseudo-classes found in the JPL AirSAR P-band image of the Landes maritime pine forest (figure 3). In general, the Tobit models fit their class-conditional histograms well. When a pseudo-map is generated from these models and subsequently converted into a true ground cover map (using the procedure described in the next

section) figure 6 results. My initial comparison of this map with ground truth indicates an accuracy on the order of 80-90% for those areas in which ground truth is known, but the confidence bounds on this estimate are large since the amount of ground truth for the Landes site is limited. This result and the others like it suggest that sarMapper's use of the eight-bit/real representation and Tobit models is sufficient to generate high-accuracy maps in near real-time. Research plans for this calendar year include quantitative evaluation of this hypothesis.

2.5 The Tobit Parametric Classifier & Map Generation

The Tobit parametric classifier is nothing more than a set of Tobit models — one for each ground cover pseudo-class identified by the human user. The classifier generates a pseudo-map for the SAR site by computing the class-conditional log likelihoods for all the pixels in each $n \times n$ pixel region in the site (these regions are contiguous, and the user specifies the map resolution n): the class of the Tobit model with the largest log-likelihood is chosen as the class for the region. This is standard maximum-likelihood parametric classification.

The resulting map is a pseudo-map in that the human user is assumed to have no prior information about the site's true ground cover. If ground truth exists (or can be inferred), the human user cross matches areas with known ground cover to the corresponding ground cover pseudo-classes. For cases in which the correspondence is one-to-one, the user can convert the pseudonym to the true ground cover name by simply re-typing the class name. For cases in which a pseudo-class represents more than one ground cover class, the user replaces the pseudonym with a list of all the true ground covers that the pseudo-class represents. Map colors can also be changed interactively.

In my experiments, I generated a pseudo map and saved the labeled pseudo-class regions. When I obtained ground truth for the site, I loaded in the pseudo-map labeled regions and changed the names/colors to reflect the truth. I then saved this ground truth data in a second label file and used it to re-generate the ground cover map — this time with the correct ground cover names and, perhaps, more meaningful colors. Representative

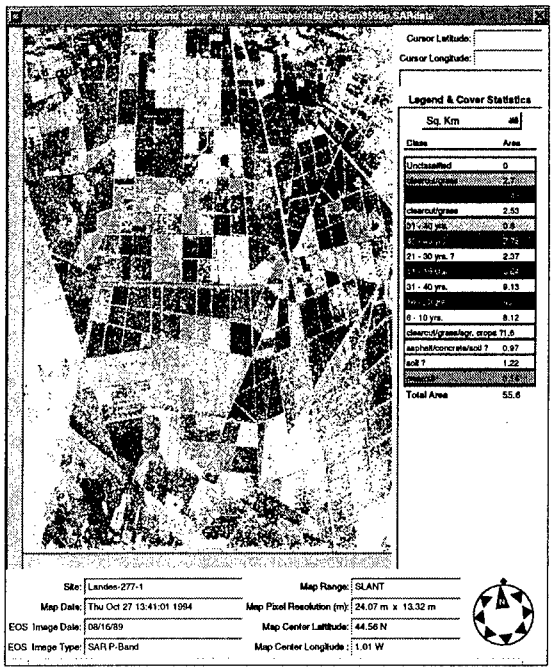


Figure 6: A ground cover map generated from the fully polarimetric image in figure 3. Fourteen different ground cover classes are distinguishable; of these seven represent the same species of tree at various ages, and four represent clearcut areas in various stages of regrowth. A full-color copy of this map can be found on the author's IU website (see references). Man-made ground cover is denoted by white and light gray in the lower/left part of this slant-range map.

maps are discussed further in section 3.

2.6 Efficiency Of The Tobit Maximum-Likelihood Classifier

Tobit maximum-likelihood mapping is both statistically efficient [Hampshire-92] *and* computationally efficient. Single-polarization JPL AirSAR images generally contain about 1.3 million pixels, so a 3-plane sarMapper color composite image (HH-HV-VV polarizations) contains about 4 million pixels. Using a 32-bit representation for each of the three polarizations, we would require almost 32 megabytes (Mb) of storage for each composite image. The requirement is 32 Mb rather than 16 because each data point would be a complex number. Let's assume that there are 10 ground cover pseudo-classes associated with an image. Using the 32-bit floating point representation, we would need

to evaluate *40 million* log-likelihood functions — one for each polarization of each pixel, given each ground cover pseudo class — to generate a high-resolution map.

By using an 8-bit envelope (real) representation for each of the three image planes, we require only 4 Mb of storage per composite image. Moreover, if we have 10 ground cover pseudo-classes, we need only evaluate $3 \times 2^8 \times 10 = 7{,}680$ unique log-likelihoods — one for each polarization of each possible pixel value, given each ground cover pseudo-class. sarMapper *pre-computes* these log-likelihoods and writes them into a look-up table prior to generating the map: when it actually generates the map, it does table look-up to get the log-likelihoods (which is about four orders of magnitude faster than actually computing them). This is why sarMapper can generate a high-resolution map with 15 ground cover pseudo-classes in about one minute on a current-technology laptop computer.

Again, the Tobit parametric estimation procedure enables sarMapper to exploit the computational efficiency of the low-complexity representation with little or no reduction in map accuracy. In fact, map accuracy might even be *better* using the eight-bit/real representation instead of the 32-bit/complex one. This possibility exists owing to the Bellman's curse of dimensionality argument in section 2.1. Research plans for this calendar year include quantitative evaluation of this possibility.

3 Preliminary Results

Figures 8 and 9 are sarMapper-generated maps of NASA's Raco, Michigan supersite. They were generated from the JPL AirSAR C-band image in figure 7, using only the HH-polarization image (not shown) in order to approximate the process of generating maps from single-polarization X-band military imagery.

I spent about five minutes studying the image in figure 7, selecting regions of vegetation, water, and tarmac, which I inferred from the image without any ground truth. I generated the low-resolution map of figure 8 in approximately five seconds, and the high-resolution map of figure 9 in approximately 15 seconds. These two maps illustrate how sarMapper might be used to focus the attention of image analysts searching for

Figure 7: An HH-HV-VV composite C-band JPL AirSAR image of NASA's Raco, Michigan supersite (shown here in grayscale — see the author's IU website for a color version). Note the airfield in the lower right quadrant of the image.

man-made ground cover of potential tactical interest: figure 8 represents a first-stage focusing process that identifies ground cover that could be either water or tarmac. This very coarse map identifies a small number of regions in the map that warrant further analysis. Figure 9 is a higher-resolution of the site. Imagine that figure 8 is used as a focus-of-attention (FOA) mask overlayed on figure 9. Detailed mapping and analysis of the masked regions (figure 9) would reveal the area containing the airfield with a small number of additional "false tarmac" areas. A simple automatic target recognition (ATR) post-processing of this image would recognize the "false tarmac" regions as water, and the airfield as the single area of potential tactical interest.

Figure 10 shows another map of the Raco, Michigan super site, produced from a longer-wavelength JPL AirSAR P-band image. Knowing nothing about the site, I generated a pseudo-labeled HH-HV-VV color composite image and sent it to Leland Pierce (U.

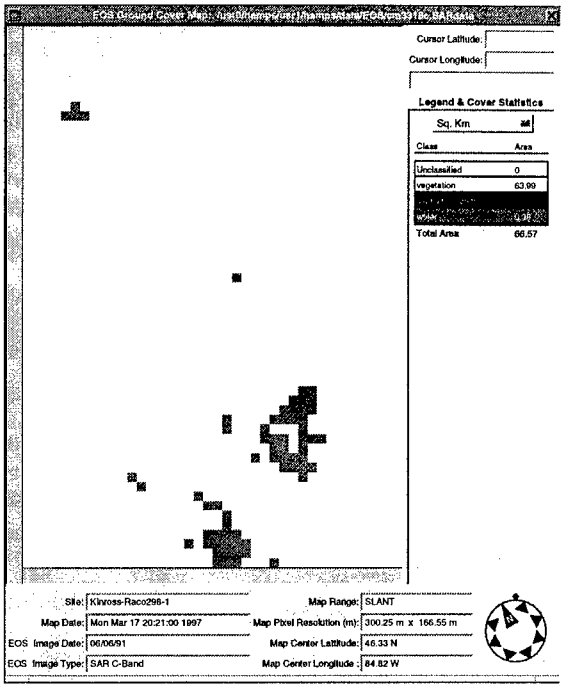


Figure 8: A coarse(300 m) resolution focus-of-attention map of the Raco, Michigan site, generate from a C-band HH-polarized image. Darker regions on the map indicate areas of potential interest (water or tarmac; the latter is of tactical interest). High-resolution mapping and Automatic Target Recognition can be focussed on these areas.

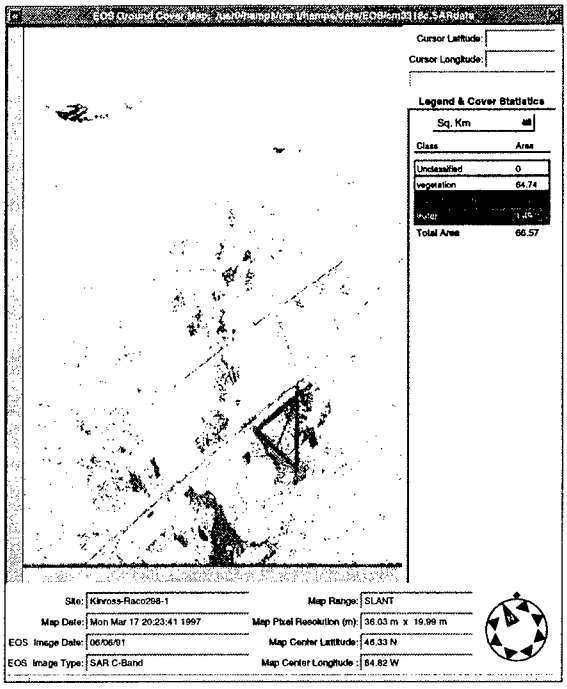


Figure 9: A fine(36 m) resolution map of the Raco, Michigan site, generate from the same C-band HH-polarized image used to generate the focus-of-attention map in figure 8. The darkest regions on the map indicate areas likely to be tarmac. The physical structure of the airfield, combined with its tarmac ground cover identify it as a legitimate target of interest.

Michigan), who graciously sent me ground truth corresponding to the regions I had sent him. I compared his ground truth to the map I had generated from the same labels and derived the true ground cover classes (shown) from the original pseudo-classes (see the author's IU web site for a color version of this map). The map is generally in good agreement with the ground truth, with some notable exceptions I discuss below. Five pseudo-classes corresponded to clearcut/grassland, two to non-scattering ground cover (water and man-made materials), two to hardwoods (aspen, birch, and northern hardwoods), and four to conifers. The P-band image does not discriminate non-scattering ground cover (water and man-made materials) from scattering ground cover well. This is due in part to the steep SAR incidence angle (small look angle) at the top of the image, which generates strong HH backscatter from all ground cover. Also, the long wavelength generates some backscatter from undulations in water surfaces.

The shorter wavelength in figure 7 generated substantially less backscatter from the water and man-made materials, so they were easier to distinguish from vegetation than they were in the P-band image. Water and asphalt/concrete should be more distinguishable in an X-band image. Nevertheless, figure 10 illustrates the utility of longer-wavelength radar for generating detailed wide-area tactical maps. A military commander planning an assault on the airfield in this image would know where to conduct paratrooper air-drops (in the fresh clear-cut areas to the north of the airfield), and would realize that an armored assault landed on the beach at the top of the map should *not* take a direct route towards the airfield. Such a route would require the armor column to penetrate a forest of old-growth hardwood trees (more than 20 meters tall). A less direct route eastward across the top of the map, and then south-southwest to the airfield would be better, since it would be through groves of aspen,

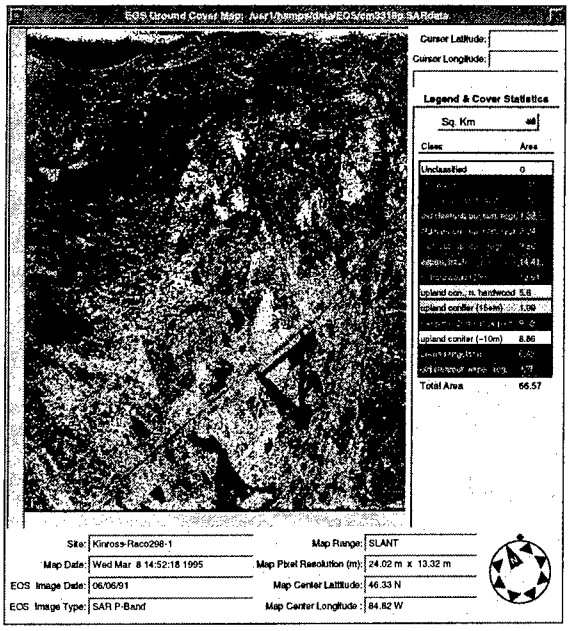


Figure 10: A detailed vegetation map of the Raco, Michigan site generated from a long-wavelength JPL AirSAR P-band image (see the author's IU web site for a color version of this map). The map indicates that sarMapper can distinguish subtle differences in ground cover, providing important tactical information to a commander who might, for example, be planning an assault on the airfield in the lower-right quadrant of the map (see text).

birch, and small conifers, which would be easier to penetrate.

4 Future Work

Pre-processing will be added to sarMapper so that it can map imagery from these three sensors (in addition to the JPL AirSAR platform):

- Lincoln Lab ADTS SAR (X-band)
- MSTAR SAR (Sandia/DOE X-band)
- NASA SIR-C/X-SAR (L, C, X-band)

Four research questions will be addressed this year. These questions — all of them technical in nature — touch on the speed, accuracy, and robustness of sarMapper's mapping and focus-of-attention capabilities:

- Can high-accuracy maps be generated from 8-bit X-band backscatter envelope data? Is the accuracy of these maps significantly better or worse than the accuracy of maps generated from the same imagery using standard maximum-likelihood techniques on floating-point representations of *complex* RF backscatter?
- Is semi-supervised learning a statistically consistent paradigm?
- Can man-made ground cover be identified consistently in SAR imagery and pseudo-maps without ground truth?
- How can a robust focus-of-attention algorithm be derived for real-time laptop implementation?

4.1 Evaluation

Pursuant to research efforts to address them, the research questions listed in the previous section will be answered by objective evaluation of sarMapper using SAR data from a wide variety of sensors.

Speed: Average semi-supervised learning time will be assessed with human-computer timing trials. Map generation times will be tabulated for a corpus of evaluation images.

Accuracy: Map accuracy will be assessed using imagery for which ground truth is known or can be inferred. Accuracy will be quoted according to general methods of statistical inference/pattern recognition, with 95% confidence bounds and ground cover confusion matrices derived from test images (or test areas within an image) not used during semi-supervised learning.

FOA: sarMapper's focus-of-attention algorithm will be assessed according to general methods for evaluating detection algorithms; namely, receiver operator characteristic (ROC) curves will be generated and evaluated for FOA performed on test imagery/maps not used for semi-supervised learning.

Acknowledgements

Some funding for this proof-of-concept research was provided by NASA in 1994. Victoria Gor (JPL) translated my original Tobit estimation source code from FORTRAN into ANSI C. Dr. Cynthia Williams (Institute of Northern Forestry) has graciously volunteered some of her time to evaluate sarMapper's potential as a useful ecological mapping and analysis tool. Her insights have proven useful to the DARPA thrust

of this research.

Eric Rignot and Pascale DuBois of JPL section 334 provided me with FORTRAN source code and the technical details of JPL AirSAR data compression/decompression algorithms[1], which I converted to ANSI-C and subsequently modified. They and Jakob Van Zyl also provided me with a number of AirSAR images and technical papers they have written on the subjects of SAR and SAR remote sensing. Leland Pierce (University of Michigan) provided me with ground truth for the Raco, Michigan super site, as well as a number of helpful insights. Thuy LeToan (University Paul Sabatier, Toulouse, France) provided me with ground truth for the Landes maritime pine forest in France. Tony Freeman (JPL, section 334) provided me with AirSAR images and ground truth for the Flevoland site in Holland.

References

The author's DARPA IU website can be reached at

http://gussolomon.ius.cs.cmu.edu/hamps/IU/index.html

[Abramowitz-70] M. Abramowitz and I. Stegun, editors. *Handbook of Mathematical Functions*, volume 55 of *U.S. Dept. of Commerce, National Bureau of Standards, Applied Mathematics Series*. U.S. Government Printing Office, Washington, D.C., 1970. Ninth printing.

[AirSAR-90] *Airsar Bulletin*. Jet Propulsion Laboratory, Pasadena, CA, February 22 1990.

[Aitcheson-66] J. Aitcheson and J. .A. C. Brown. *The Lognormal Distribution*. Cambridge University Press, London, 1966.

[Amemiya-85] T. Amemiya. *Advanced Econometrics*. Harvard University Press, Cambridge, MA, 1985.

[Beckmann-67] P. Beckmann. *Probability in Communication Engineering*. Harcourt Brace and World, New York, 1967.

[Bellman-61] R. E. Bellman. *Adaptive Control Processes*. Princeton University Press, Princeton, NJ, 1961.

[Cramér-46] H. Cramér. *Mathematical Methods of Statistics*. Princeton University Press, Princeton, NJ, 1946.

[Davenport-58] W. B. Davenport, Jr. and W. L. Root. *An Introduction to the Theory of Random Signals and Noise*. McGraw Hill, New York, NY, 1958. *Re-printed by the IEEE press in 1987, ISBN 0-87942-235-1.*

[DuBois-87] P. DuBois et al. "Data volume reduction for imaging radar polarimeter" [sic]. In *Proceedings of IGARSS-87*, 1987. Abridge version in *NASA Tech Briefs*, Vol. 13, No.2, item 37.

[Duda-73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY, 1973.

[Hald-49] A. Hald. "Maximum Likelihood Estimation of the Parameters of a Normal Distribution which is Truncated at a Known Point." *Skandinavinsk Aktuarietidskrift*, 32:119–134, 1949.

[Hampshire-88] J. B. Hampshire II. *A Non-Rayleigh Model for Ultrasonic Backscatter in Myocardium*. Master's thesis, Theyer School of Engineering, Dartmouth College, August 1988.

[Hampshire-92] J. B. Hampshire II and J. W. Strohbehn. "Tobit maximum-likelihood estimation for stochastic time series affected by receiver saturation." IEEE Transactions on Information Theory, IT-38(2):457–469, March 1992.

[Hampshire-93a] J. B. Hampshire II. *A Differential Theory of Learning for Efficient Statistical Pattern Recognition*. PhD thesis, Carnegie Mellon University, September 1993.

[Hampshire-93b] J. B. Hampshire II and B. V. K. Vijaya Kumar. "Differentially Generated Neural Network Classifiers are Efficient." In C. A. Kamm, G. M. Kuhn, B. Yoon, R. Chellappa, and S. Y. Kung, editors, *Neural Networks for Signal Processing III*: Proceedings of the 1993 IEEE Workshop, pages 151–160, New York, September 1993. The Institute of Electrical and Electronic Engineers, Inc.

[Ishimaru-78] A. Ishimaru. *Wave Propagation and Scattering*, vol. 1. Academic Press, New York, NY, 1978.

[1] These JPL AirSAR compression/decompression algorithms are unrelated to the algorithms described in section 2 (which relate to 8-bit displayable representations of SAR imagery); instead they relate to efficient storage of the Stokes matrix coefficients used as the fully-polarimetric AirSAR data representation.

[Kong-87] J. A. Kong, A. A. Swartz, H. A. Yueh, L. M. Novak, and R. T. Shin. "Identification of Terrain Cover Using the Optimum Polarimetric Classifier." *Journal of Electromagnetic Waves and Applications*, 2(2):171–194, 1987.

[Nakagami-60] M. Nakagami. "The m-distribution — a general formula of intensity distribution for rapid fading." In W. C. Hoffmann, editor, *Statistical Methods in Radio-Wave Propagation*, chapter 1, pages 3–36. Pergamon Press, New York, 1960. *A summary of the author's work, contemporary with that of S. O. Rice, but relatively unknown in the U.S. until this publication.*

[Rao-45] C. R. Rao. "Information and accuracy attainable in the estimation of statistical parameters." *Bulletin of the Calcutta Mathematical Society*, 37:81–91, 1945.

[Rice-44] S. O. Rice. "Mathematical Analysis of Random Noise." *Bell System Technical Journal*, 23:283–332, 1944. *First of two parts.*

[Strohbehn-75] J. W. Strohbehn, T. Wang, and J. P. Speck. "On the Statistics of Line-of-Site Fluctuations of Optical Signals." *Radio Science*, 10:59–70, January 1975.

[Strutt-29] J. W. Strutt. *The Theory of Sound.* MacMillan Co., London, 1929. *The author is better known as Lord Rayleigh.*

[Tobin-58] J. Tobin. "Estimation of Relationships for Limited Dependent Variables." *Econometrica*, 41:24–36, 1958.

[Tukey-77] J. W. Tukey. *Exploratory Data Analysis.* Addison-Wesley, Reading, MA, 1977.

Combining Geometric and Appearance Models for Change Detection

Anthony Hoogs
University of Pennsylvania,
Lockheed Martin Management and Data Systems
P.O. Box 8048
Philadelphia, PA 19101
hoogs@mds.lmco.com

Abstract

A method for change detection using a combination of view-based and model-based representations is presented. Coarse geometric models are enhanced with appearance-based information using a small number of training images, resulting in a hybrid representation that maps local appearance characteristics onto object subparts. This is accomplished by matching segmented features in a training set to model edges, and generalizing this information over multiple images using evidential combination. The hybrid representation provides a more realistic prediction of object appearance than geometry alone, thereby improving the performance of change detection and pose adjustment. The learning behavior of the system is studied as a function of the number and characteristics of the images in the training set. Results are presented showing the level of improvement of system performance in a change detection task as images are added to the training set, in comparison with using a purely geometric approach.

1 Introduction

Recently view-based object recognition systems have demonstrated significant capabilities in recognizing complex 3D objects in simple scenes [6, 9, 2]. These systems operate by learning appearance characteristics of objects from training imagery without recovering geometry. This data-driven approach allows view-based systems to learn visually complex objects with many surface features, but often requires an extensive training set spanning the range of all parameters affecting object appearance. In effect, the segmentation problem is circumvented by enumerating many possible imaging conditions in advance. Also, figure-ground discrimination and occlusions can cause difficulties because features are computed over the complete object or image.

In previous work [4, 5] we describe an alternative approach to object representation, called *segmentation modeling*, that uses learning from training images, but is also model-based. The learning paradigm is designed to incrementally improve its performance as training images are added, beginning with no training images at all. The system also relies on coarse 3D geometric models of objects, such as CAD models, but the objects themselves may have complex surface features that would cause difficulties for most model-based methods because the projected model does not effectively match the objects' true appearance. The geometric model provides a spatial framework for localizing these surface appearance characteristics learned from training data, and it also allows the system

to operate successfully with very little training data.

As more training data is given to the system, the accuracy of the model should improve. To measure this quantitatively, the hybrid representation has been incorporated into a larger change detection system that attempts to identify significant changes in a scene over time. Images of the scene are taken periodically, from arbitrary viewpoints and with arbitrary illumination angles. Currently, the change detection system is focused on detecting the disappearance or removal of objects from the scene. Objects of interest are modeled with coarse CAD models (usually by hand), and these are used to detect the objects in later images. The hybrid model representation is well-suited to this domain, since images are input only periodically, and the wide range of imaging conditions precludes the use of pixel-level comparison. Results showing how our representation improves performance are discussed in Section 4.

Combining model-based and view-based representations has significant advantages over either method alone, especially when the scene is complex. Using only geometry to predict object appearance leads to significant performance degradation when scene or imaging conditions give rise to poor segmentations. However, these segmentation behaviors caused by sensor imaging effects, surface albedoes, and unmodeled surface features are accounted for implicitly through the training imagery. Furthermore, these appearance characteristics are localized to specific object features. Other recent work [7, 2, 10] has concentrated on localization, since this allows geometric indexing, robustness under occlusion and figure-ground discrimination. However, Ikeuchi's system is specialized to range data, and the other systems do not use 3D models, leading to the need for prototypical views and relatively large training sets. By using the 3D model to provide geometric constraints, we avoid these difficulties under many circumstances.

The formulation of segmentation models is briefly reviewed in the next section. Section 3 describes the change detection system, and how segmentation models are incorporated into it. Section 4 presents results and analysis comparing change detection with and without segmentation models created over a range of image training sets. The results indicate that segmentation models improve change detection performance, particularly on difficult problems.

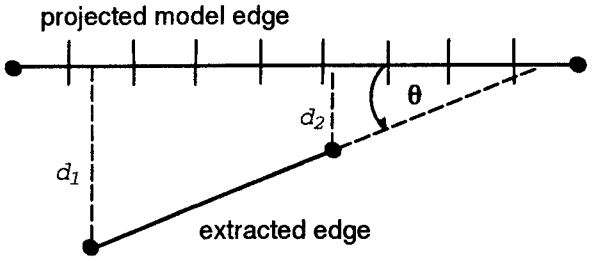


Figure 1: The correlation of an extracted edge to a projected model edge.

2 Segmentation Models

In previous work [5, 4], we describe an approach to modeling segmentations of geometric representations. This section summarizes that work, examining the specific case of segmentations of the edges of polyhedral object representations.

Segmentation models are constructed by associating segmentation information from a training set with projected geometric features. This requires that geometric models of objects in the scene are available, and that the images used in construction are registered (calibrated) to the scene.

When a registered image is presented to the system, the first stage of processing is to extract salient lines from the image. We use a line finder built upon the Canny edge detector [1, 8] to extract segmentation features. The edges of modeled objects are projected onto the image, and a linear correlation technique is applied to associate projected object edges and extracted segmentation edges based on distance and orientation, as shown in Figure 1. If $d_1 + d_2 < D_T$ and $\theta < \theta_T$, where D_T and θ_T are fixed thresholds, then the extracted edge is determined to be correlated to the object edge.

The correlated edges are then used to update the segmentation models of the model edges. Each model edge has its own Object Edge Segmentation Model (OESM). This may be thought of as a statistical generalization of observed edge characteristics, or *attributes*, along an object edge. Information computed from image segmentations is mapped into the OESM according to the projection of the object edge into the image. Currently, the attributes included in the segmentation models are *correlation* and *gradient magnitude*. The correlation attribute is the proportion of the object edge covered by observed edges, and the gradient is measured by taking the difference of the average intensities of rectangular regions on either side of the edge.

The segmentation model is defined in terms of edge attributes. For the purposes of this paper, we assume that viewpoint and illumination orientation, termed the *imaging parameters*, are the dominant factors causing differences in segmentations. Each segmentation model is defined as a mapping M from imaging parameters into segmentation values for an object edge. If the segmentation values are represented as a vector $\mathbf{A}$ of feature attributes (or distributions over each attribute), then the mapping M is defined as $M : \mathbf{S} \times \mathbf{V} \rightarrow \mathbf{A}$ where $\mathbf{V}$ and $\mathbf{S}$ are imaging parameters (viewpoint and solar orientation) affecting the segmentation. In the case of the OESM, $\mathbf{A}$ is defined to be $(c, g)^T$ where c is the correlation attribute and g is the gradient attribute. $\mathbf{S}$ is a vector, described by two angles θ and ϕ, and $\mathbf{V}$ is also a vector, described by (α_v, β_v).

The OESM may be simplified by reducing the partitioning of the $(\mathbf{S}, \mathbf{V})$ space based on whether the edge is interior or occluding relative to viewpoint and illumination [4], as shown in Figure 2. For each edge, this leads to twelve partitions of the $(\mathbf{S}, \mathbf{V})$ space, or *imaging modes*; 3 possibilities for viewpoint (invisible edges are ignored), and 4 for illumination. Separate models are maintained for each mode.

The attributes computed from observed edges are associated to the corresponding object edge by partitioning the object edge into equally-sized discrete intervals, each containing its own attribute vector. This provides spatial localization of attribute values within an object edge.

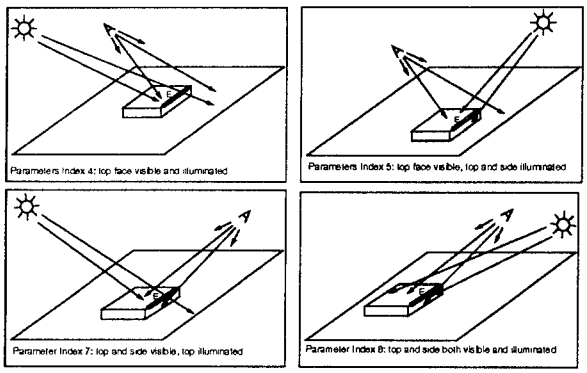


Figure 2: Prototypical imaging conditions for four common imaging modes. The edge of interest is enhanced and labeled E.

Segmentation models are learned over a set of training imagery containing representative views and scene conditions. The technique used to build segmentation models should perform well on small sets of training images and should allow incremental updates as new images are acquired. Using statistical methods for developing segmentation models has the advantage that segmentation errors and spurious, atypical segmentations tend to be filtered out over a set of imagery. Because segmentation information is combined across multiple images, there is very little dependence on specific parameter values used in segmentations – the same parameter values can be used on every image in the training set. Also, dynamic construction allows objects with arbitrarily complex surface features to be modeled, providing robustness on images of real buildings in outdoor settings.

Given an image I taken from viewpoint $\mathbf{V}_I$ with a single illumination source at orientation $\mathbf{S}_I$, the segmentation model M for feature f is updated as follows. Let A_M be the representative attribute vector corresponding to $\mathbf{V}_I$ and $\mathbf{S}_I$. The attribute vector A_I for image I is computed, and used to update A_M:

$$A_M = \Delta(A_M, A_I)$$

where $\Delta(A_M, A_I)$ is a function defined by the

learning paradigm. If I is the first image in the training set, then A_M is initialized to A_I.

The Δ function used to update the OESM is a simple Bayesian estimator (with uniform prior),

$$A_M = \frac{i}{i+1}A_M + \frac{1}{i+1}A_I$$

where i is the number of images used to build A_M, not including the new image I. This function was chosen over other Δ functions for a number of reasons. A Kalman filter approach is sensitive to the initial image used to create the segmentation model, giving it an inappropriate importance. Over time this effect disappears, but within the data sets we used (40 images) the influence of a poor initial image was still apparent. The median was also tested, but it proved to be less informative because the correlation attribute tends to be exactly zero or one in most edge intervals derived from I. Thus the correlation profiles contained mostly zeroes and ones, which do not provide accurate estimates of probabilities.

The complete process for building segmentation models on object edges is outlined in Figure 3. The *combined correlation* attribute is computed using the geometry of the extracted edges. When multiple segmentation edges are correlated to a common object edge, their correlation intervals may overlap, or there may be gaps in coverage of the object edge. The combined correlation attribute is computed by merging the correlation intervals of all of the extracted edges correlated to one object edge. Similarly, the combined gradient attribute is computed by taking the correlation-weighted average of the gradient values of any overlapping extracted edges.

1. For each image I in T, perform line extraction yielding a set of extracted segments S_x.
2. Determine the imaging mode q based on the viewpoint and illumination of I.
3. For each object edge E_o, find the set of segments S_o, $S_o \subseteq S_x$, that are correlated with the projection of E_o in the image plane of I.
4. For each segment in S_o, compute the gradient attribute.
5. For each interval i in E_o, compute the combined correlation attribute c_i across S_o.
6. Using the combined correlation attribute, for each i compute the combined gradient attribute g_i across S_o.
7. For each interval i in E_o, update each attribute in mode q of E_o using the Δ function:

$$E_{o,q}(i,c) = \frac{N}{N+1}E_{o,q}(i,c) + \frac{1}{N+1}c_i$$

$$E_{o,q}(i,g) = \frac{N}{N+1}E_{o,q}(i,g) + \frac{1}{N+1}g_i$$

where $E_{o,q}(i,g)$ is the value of the i^{th} interval of the gradient attribute of edge E_o in mode q.

Figure 3: The steps in constructing edge segmentation models. The training set T contains m images showing the same object(s) without physical change, but from a variety of viewpoints and illumination conditions.

3 Using Segmentation Models

Segmentation models provide a formulation for estimating segmentation behavior based on previous imagery. These estimates can then be used by higher-level systems to derive predictions of object appearance in new images that are more accurate than using only object geometry. Geometry in combination with prior segmentation information should lead to improved performance in many model-based vision algorithms [10].

To investigate how segmentation models could be used effectively to improve the performance of higher-level vision systems, we have developed a change detection system that identifies changes in time-sequenced images (not video) of man-made structures such as buildings, roads, construction areas, etc. in outdoor scenes. In this scenario, images of a fixed location are taken from aerial sensors over a period of time. Buildings and structures may be created, destroyed or modified; more frequently, the appearance of structures varies considerably based on weather, season and imaging parame-

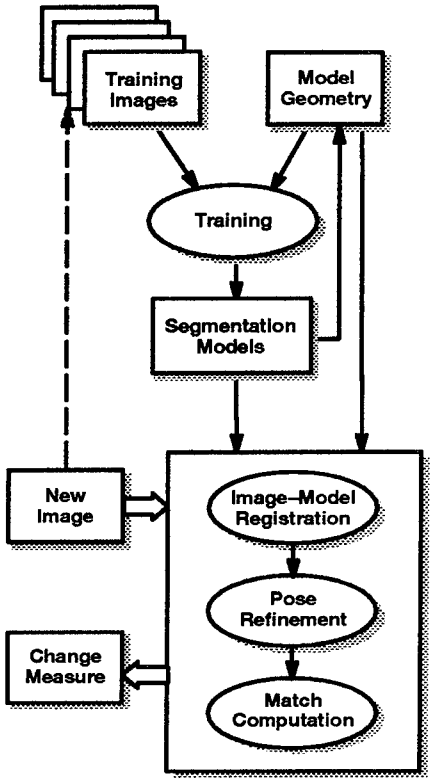


Figure 4: Change detection process flow.

ters. The goal of our system is to identify true changes in structures while ruling out apparent differences due to non-geometric effects.

The change detection system architecture is shown in Figure 4. Initially, segmentation models are created using the model geometry of structures in a scene and registered (calibrated) training images of the scene. When a new image of the scene is presented to the system, it is registered to the geometric models or other scene features using manual or automatic techniques. Image registration itself is a well-studied area, and is beyond the scope of this work.

Many existing image registration algorithms, manual or otherwise, typically result in at least 2 pixels of error because of unmodeled sensor distortions and noise. To compensate for this, the system performs a local 2D translational pose refinement step that adjusts the position of a single object model with respect to the image. Described in previous work [5, 3], the pose adjustment step uses a hierarchical approach based on relaxation of matching tolerances between the image and the object model. A bounded search area is defined locally around the initial object position, and this area is scanned in the image at increasing levels of spatial resolution and decreasing levels of match tolerance. At each level a fixed number of the points in the area yielding the highest match scores are kept, and are explored at the next level of resolution. The object position corresponding to the highest match score at the finest spatial resolution (usually one pixel) is returned.

The next stage of the algorithm, match computation, is a virtual operation; the pose refinement stage actually computes the match score that is the output change measure. The same matching algorithm or metric between image and object model is used in pose refinement and change detection, so that the optimal value found in pose refinement is considered to be the best estimate for the change measure.

The match metric is computed using the segmentation model. Our previous work demonstrated the utility of using the segmentation model over pure model geometry for pose adjustment [5], and we have continued to develop this approach.

The model matching procedure is similar to that used in segmentation model construction. An image of the scene is segmented and the appropriate modes of each object edge (i.e., which segmentation model to use) are computed based on prior knowledge of the viewpoint and illumination directions. Since the search does not compute object rotations, these modes remain constant throughout the search. The segmented lines are correlated with the projected model edges for a given model position. Edge attributes are computed from the image for each visible model edge (the edge *profile*), and a match score between the single-image attributes and the prior edge segmentation models is calculated. The match score is then used to guide the hierarchical search by ranking position hypotheses.

Defined over an object model with multiple edges, the match metric μ_O has three primary components:

$$\mu_O = \frac{\sum_{i=1}^{K_G} l_i}{\sum_{i=1}^{K+K_G} l_i}\mu_G + \frac{\sum_{i=1}^{K} l_i}{\sum_{i=1}^{K+K_G} l_i}(\mu_c\mu_g)^{1/2}$$

μ_G is purely geometric term that accounts for object edges that do not have any training data in the current imaging mode. It is the length-weighted sum of the geometric match scores of its visible edges:

$$\mu_G = \frac{\sum_{i=1}^{K_G} l_i m_i}{\sum_{i=1}^{K_G} l_i}$$

where K_G is the number of visible object edges without training data, l_i is the length of the projection of object edge i, and m_i is the proportion of edge i that is correlated to any extracted edge.

The second term involves μ_c and μ_g, which are derived from the two segmentation model attributes, correlation and gradient. μ_c is defined over all edges of the object:

$$\mu_c = \frac{\sum_{j=1}^{K} l_j \sum_{i=1}^{n} \min(P_j(i), S_j(i))}{\sum_{j=1}^{K} l_j \sum_{i=1}^{n} P_j(i)}$$

where K is the number of visible object edges with training data, l_j is the length of the projection of object edge j, n is the number of intervals in the edge model, $P_j(i)$ is the correlation value of the i^{th} interval of the edge model of j, and $S_j(i)$ is the correlation value of the i^{th} interval of the profile of j extracted from the new image.

A graphic representation of μ_c is shown in Figure 5 for a single object edge. μ_c is actually computed using all model edges visible in the image to approximate a joint density function between the intervals of all object edges.

μ_g is computed using the direction of the gradient attribute. The gradient direction of an object edge encodes the relative albedoes of the surfaces bordering the edge, and can often be consistently extracted from images in the same mode. μ_g is defined to be the ratio of the number A of correlated intervals that have the correct gradient direction to the number B of all correlated intervals:

$$\mu_g = \frac{\sum_{j=1}^{K} l_j \frac{A_j}{B_j}}{\sum_{j=1}^{K} l_j}$$

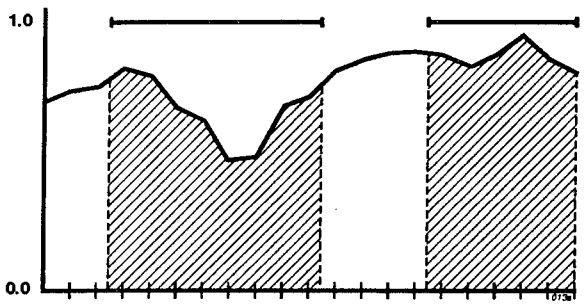


Figure 5: A graphic depiction of the correlation match metric μ_c. The horizontal axis represents the segmentation model intervals along the object edge, and the vertical axis represents the probability of an interval being correlated to a segmentation feature. An example extracted profile from one image might consist of the top straight segments, corresponding to two segmented lines correlated to the object edge. μ_c is defined to be the ratio of the areas of the shaded regions to the area under the entire model profile curve.

The complete metric μ_O is the weighted sum of the two terms. The weighting function is the length of the object edges that do (μ_c and μ_g) or do not (μ_G) have training data compared to the total length of all object edges. The attribute metrics μ_c and μ_g are multipled together because μ_g is dependent on μ_c; only intervals contributing to μ_c are used to compute μ_g. The square root of the attribute term is used to normalize it with μ_G.

4 Learning Analysis

Every learning-based system is dependent on its training data. However, by combining training images with known geometry we hope to reduce the burden of each representation while retaining their advantages. We desire to produce a system that performs well on very small training sets that are highly unconstrained, i.e. no assumptions are made about the distribution of

viewpoints or illumination angles in the training images.

To assess the effectiveness of our representation, we have compared the hybrid system to one using geometry alone on the same change detection problems. The images in Figure 7 show four aerial views of an example scene containing a difficult building obscured by trees. The building has a flat white roof, but it is occluded by tall trees and their shadows. We have a data set containing 24 images of this scene, which includes a number of other buildings of varying detectability. Four more of these images are shown in Figure 8; note that in three of these images the object model is close to the image, but further pose refinement is necessary to align the model edges with the image.

Using six images for training on the displayed building model, the results of the geometric and hybrid change detection systems are plotted in Figure 6. The training images were selected temporally, as they would be in a true change detection framework; the 24 images were taken at random intervals over a period of four months, and the training images are the first six of these. Hence, there is no attempt at optimizing the training data to span the viewing sphere or to meet any other criterion. Two of the six training images are the upper right and lower left images in Figure 8.

In Figure 6, the dashed line connects the change levels computed by using geometry alone ($1 - \mu_G$), while the solid line plots the change levels computed using the segmentation model metric ($1 - \mu_O$). For the segmentation model case, the scores on the training images (1 - 6) are plotted to show the best expected performance given the training data. All images except I_{12} show the building; for I_{12} the building was shifted to another part of the scene to simulate a change in the building (I_{12} is the bottom right image in Figure 8). Note that the peak in change level at I_{12} is nearly equal for both cases, and is clearly more distinctive in the segmentation model case.

The data shows that, in this case, the training data improves performance significantly on the images without change while producing a change level nearly equal to the purely geometric model on the image showing change. Note that the absolute values of the change levels are not important – it is the separation between change and no-change image results that matter.

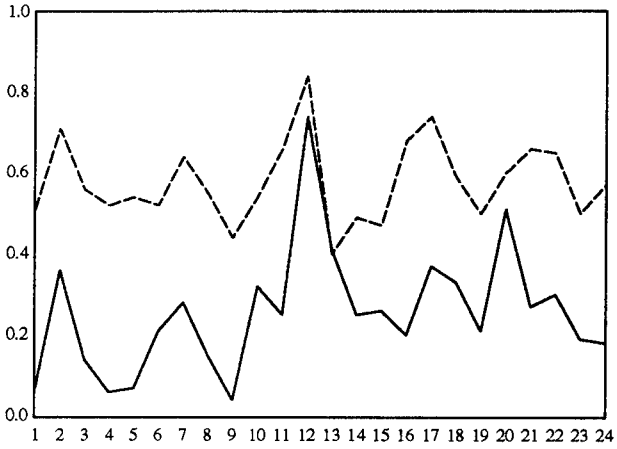


Figure 6: Image-by-image comparison of the match level over a set of images. The images are plotted on the horizontal axis; each number represents a different image I_i of the object. The vertical axis is 1.0 - μ, or the computed change level. The dashed line represents geometry only, while the solid line shows the improved performance from using the segmentation model.

To assess the system's learning capabilities on small training sets, we analyzed its performance as the number of training images varies from 0 to 10 (Figure 7 shows four of these training images). After training on the building model shown, the system was tested at each number of training images on the 14 images not included in any training set. Four of these test images are shown in Figure 8 (three of these show no change), and four more images showing change are in Figure 9. The 14 change images were "created" by shifting the building model in each of the no-change images. The building was placed by hand intentionally close to other buildings and distractors to increase the difficulty of detecting change. No system parame-

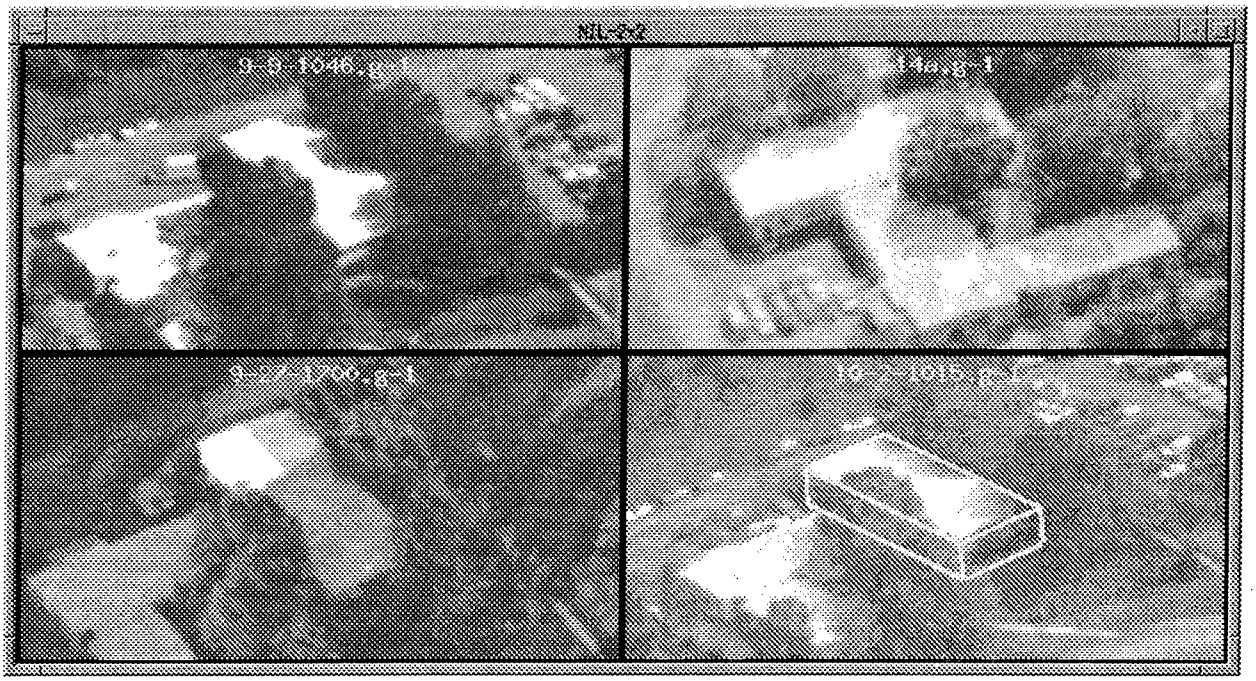


Figure 7: Four of ten training images used in computing the learning curves. The building model is shown overlaid on the image in the bottom right pane.

ters (including segmentation parameters) were adjusted specifically for or during the experiments.

The results are shown in Figure 10. This graph shows the average level of change computed for three cases – on training images, on test images showing no change, and on test images showing change – as the number of training images is increased. For the cases showing no change (respectively change), the desired score is 0 (respectively 1). Ideally, the change test curve should converge to 1 and the no-change test curve should converge to 0 as more images are added to the training set. The critical metric on the graph is the level of discrimination between change and no change, which corresponds to the separation of their curves (i.e. their ratio).

This divergence is apparent in the graph, especially as the first few training images are added. The case of 0 training images corresponds to using only model geometry; in this case, the discrimination between the change and no-change images is negligible (.05). With one training image, the difference increases to .14, and it increases to .17 at 10 training images. However, the ratio of the change to no-change values increases steadily and reaches a maximum of 1.6 at 10 training images. Thus, on this data set, the system demonstrates rapid improvement when using small numbers of training images.

As expected, the training data produces the best performance. The test images closely follow the training data, however, indicating that the segmentation model has learned characteristics of the training set that are also found in the test set. On the test images showing change, the level of change score decreases as more images are added to the training set because of the increased generality of the model. The model generality can be measured by the performance on its training data – if the level of change is nonzero, then there must be a difference between training images *in the same mode*. Thus this value measures the variance in the training data in a quantitative, task-based way that provides direct feedback on expected learning performance. This value inherently measures the difficulty of visually interpreting the scene in a

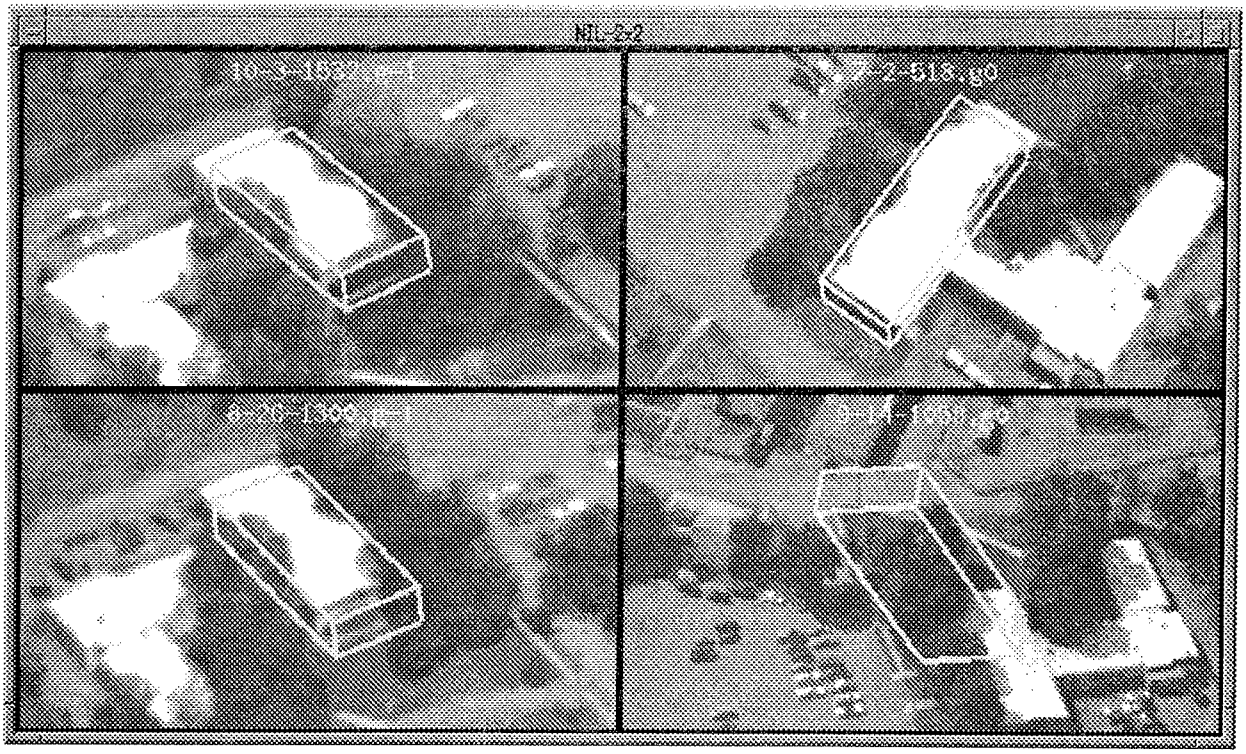


Figure 8: Three images showing no change, and one showing "simulated" change by building displacement (lower right). The building model is overlaid in white.

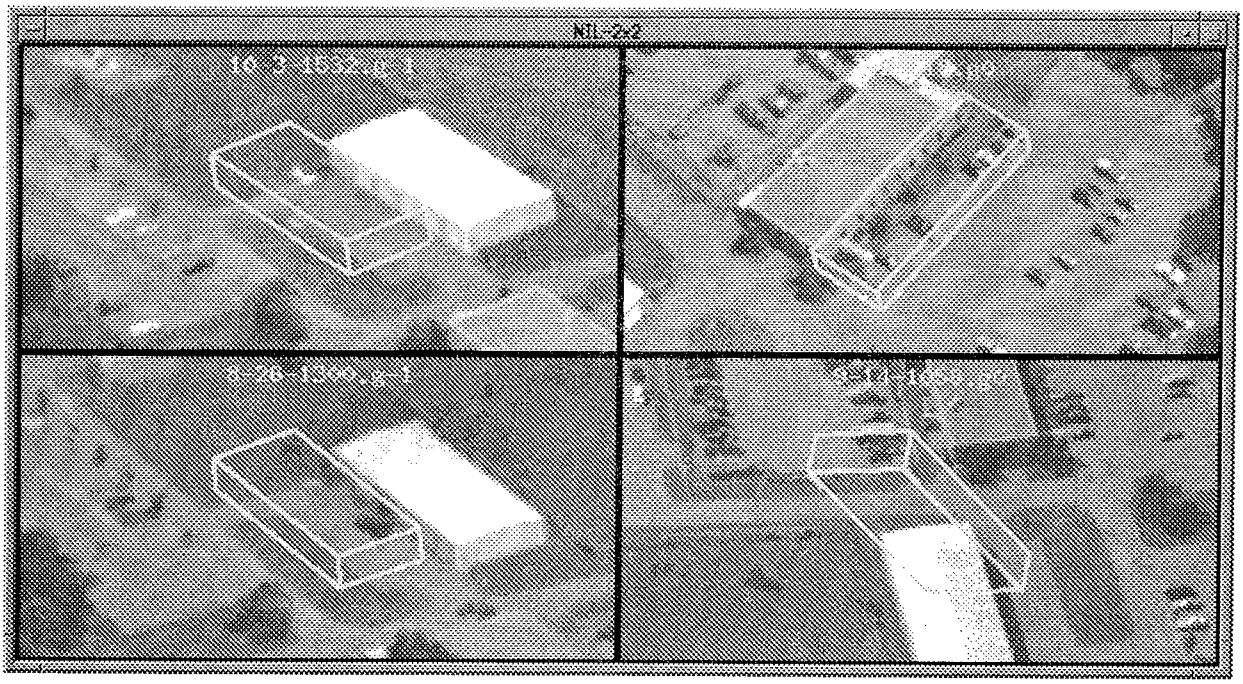


Figure 9: Four images showing change (lower right). The building model was placed near similar buildings to act as distractors.

model-based way.

As a means of assessing which imaging conditions contribute to the model variance, the learning curves were recomputed without partitioning on illumination angle. In other words, only viewpoint was used to divide the images into imaging modes. The resulting data is shown in Figure 11.

The results are surprising. The system performance actually improves when illumination is *not* used as a means of discriminating between images (the ratio of change to no-change increases to 1.8 at 10 training images). The model variance is higher, since the metric values on the training images are higher, but the change values are also higher. This phenomenon could be caused by the reduction in the total number of imaging modes. Since only viewpoint is used, there are three imaging modes, not twelve, and there is a corresponding increase in the number of images in each mode. This increases the variance of each mode, but introduces a greater range of attribute values that seem to correspond better to the test data.

It is worth mentioning that in many of the change images, the pose adjustment step aligned the building model with the distractor building, since the distractor is locally the highest match. Despite this, the system was still able to produce a low match score because the distractor building does not have the same appearance characteristics as the source building.

It is also worth noting that the variance of the level of change scores impacts system performance. In the experiments, the standard deviations of these scores typically did not overlap, i.e. the average no-change value plus one s.d. was less than the change value minus one s.d.

5 Conclusion

This paper describes how our method for integrating view-based and model-based representations is used in a higher-level system performing change detection. By mapping segmentation features from small sets of training images

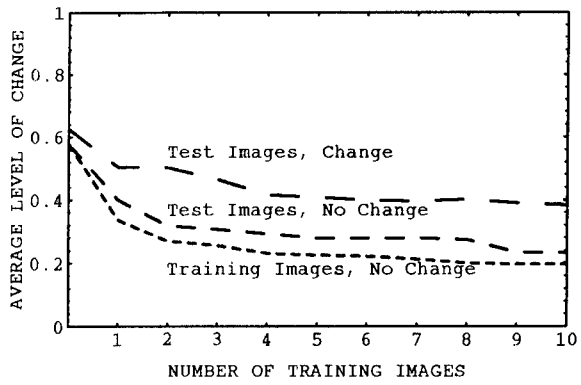


Figure 10: The learning curve using both viewpoint and illumination to partition the imaging modes.

onto 3D geometric models, the system provides data-driven prior information to improve appearance prediction. Model geometry is used to constrain the large number of parameters affecting segmentation behavior, so that performance improvement is apparent on small training sets. Segmentation models also yield a measure of the difficulty of scene interpretation, which is useful for predicting system performance.

By merging appearance characteristics and 3D geometry, the hybrid representation enhances model matching in domains where prior 3D models are available. The work presented here is necessarily limited in scope, but the techniques and principles described could be expanded and generalized to incorporate many problem domains, such as 3D object recognition and image registration.

There are many open issues in segmentation modeling that we are pursuing. The edge-based models described here can be generalized to two-dimensional, surface-based models for richer scene description. Additional edge attributes could provide useful information, and the evidential framework should be expanded to include edge, face and object information in a

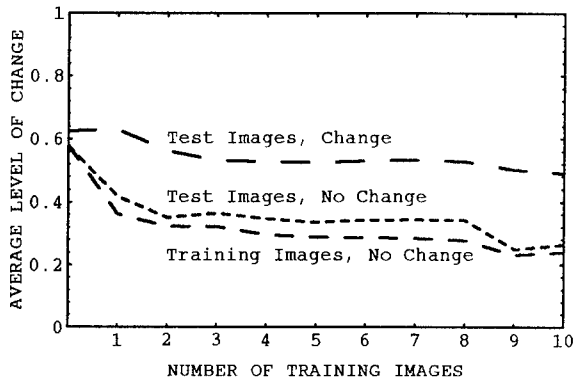


Figure 11: The learning curve using only viewpoint to partition the imaging modes.

common representation. Such a representation could then be applied to study the effects of imaging parameters, interpolation across imaging modes, and other interesting problems.

References

[1] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[2] M. Costa and L. Shapiro. Scene analysis using appearance-based models and relational indexing. *Proceedings of the International Symposium on Computer Vision*, November 1995.

[3] A. Hoogs. Pose refinement using a parameter hierarchy. *Proceedings of the ARPA IU Workshop*, February 1996.

[4] A. Hoogs and R. Bajcsy. Using scene context to model segmentations. *Proceedings of the IEEE Workshop on Context-Based Vision*, June 1995.

[5] A. Hoogs and R. Bajcsy. Model-based learning of segmentations. *Proceedings of ICPR*, AugustJune 1996.

[6] H. Murase and S. Nayar. Learning object models from appearance. *Proceedings of AAAI: Recognition*, July 1993.

[7] A. Pope and D. Lowe. Learning object recognition models from images. *Proceedings of the 4th ICCV*, May 1993.

[8] V. Venkateswar and R. Chellappa. Extraction of straight lines in aerial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:1111–1116, November 1992.

[9] J. Weng, N. Ahuja, and T. Huang. Learning recognition and segmentation of 3d objects from 2d images. *Proceedings of ICCV*, pages 121–128, 1993.

[10] M. Wheeler and K. Ikeuchi. Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3), March 1995.

Sensitivity Analysis and Learning Strategies for Context-Based Vehicle Detection Algorithms

P. Burlina V. Parameswaran R. Chellappa
Center for Automation Research, University of Maryland
College Park, MD 20742-3275

Abstract

Within the framework of the RADIUS project, the University of Maryland at College Park has been developing algorithms for the detection of prespecified vehicles in designated areas as well as the detection of global vehicle configurations in aerial imagery. These algorithms are characterized by several parameters which present problems when unsupervised batch-mode operation on a large set of images is required. In this paper, we present decision-theoretic approaches for the off-line training of most parameters. The remaining parameters are tuned using automatic calibration techniques. These techniques use control patches present in the site model which allow the derivation of empirical ROC curves from which optimal operating points are chosen. Different optimality criteria are presented. Several examples from the RADIUS dataset are provided.

1 Introduction

Context-based aerial image understanding (AIU) has been studied quite extensively recently; the approach is often referred to as "site-model-based image exploitation", *e.g.* [1; 5]. Such an approach is very well suited for routine AIU work such as detection and counting of vehicles and global vehicle configurations, because it enables discrimination between irrelevant changes (*e.g.* illumination changes, seasonal variations, etc.) and actual changes. The approach consists of maintaining an abstraction of a site, referred to as the *site model*. The site model consists of a coordinate system and various *features* which are models of the objects that the site consists of (*e.g.* parking lots, roads, buildings, etc.). A typical AIU task would be to detect changes in a newly acquired image of the site. Note that the new image may have been taken with a different illumination level or in a different season from the earlier images. Prior to doing any processing on the image, it is necessary to *register* [4] the image with the site, which means calculating the image's transformation with respect to the site's frame of reference. In other words, registration provides a mapping from features in the site, which are image-independent, to the image in question. This enables running the detection algorithms on selected portions of the image, for example, parking lots or other areas of interest. The detection algorithms, however, depend upon several parameters as input. The optimal choice of these parameters can be very image-dependent, and clearly for reasonably accurate results one cannot use the same parameters for every image. Optimal choice of these parameters appears to need extensive input from the image analyst on a per-image basis. This presents an obvious problem as regards the application of these algorithms in batch (*i.e.* unsupervised) modes over large image databases.

In this paper, we address various issues involved in solving the above problem. We consider limiting, or completely eliminating, the number of tuning parameters using the following strategy: Preliminary sensitivity analysis is used to identify those parameters to which the results of the algorithm are most sensitive. A compound measure of sensitivity is chosen as the expected risk function, computed empirically over a set of training images. The parameters to which the algorithm is least sensitive are "frozen" to their best values (off-line parameter optimization). We provide "on-line" training tools for the remaining parameters upon which the performance depends the most. For context-based image understanding systems, the availability of a site model is

The support of the Defense Advanced Research Projects Agency (ARPA Order No. 8979) and the U.S. Army Engineer Topographic Center under Contract DACA76-92-C-0024 is gratefully acknowledged.

a powerful asset for designing automatic (on-line) parameter calibration tools. For on-line parameter calibration, we exploit *control patches* present in the site model, which represent fixed areas of a given site and are used for automatic parameter tuning. This approach is explained in detail in Section 4. The approaches used for parameter training and optimization are set in a classic hypothesis-testing framework using Bayesian and Neyman-Pearson strategies.

A related issue is the assessment of the sensitivity of the algorithms to non-image-dependent parameters, for example, to model or template parameters. This problem is important for site-model-based exploitation; some results are reported in Section 6.

2 Brief Outline of Algorithms

Vehicle Detection and Counting

The aim of this module [3] is to reliably detect and count vehicles in aerial images. An example of detection in and around an intersection is shown in Figure 1 and another example on a road is shown in Figure 2.

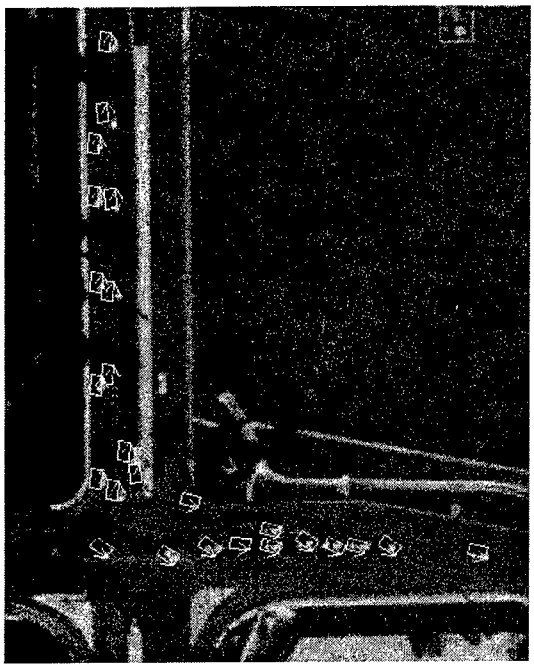


Figure 1: Vehicle detection near an intersection.

The vehicle detection process consists of two stages, an edge detection stage and a testing stage. The edge detection stage uses the Canny edge operator [2]. This stage involves specification of two thresholds, referred to as the high and low thresholds, and a mask size for convolving the image with the Canny operator. The testing phase operates on the edge data generated from the edge detection phase. A generalized Hough transform of the image is calculated using the known shape and size (user-specified) of the sample vehicle provided, and a vote for possible centers of vehicles is thus calculated for each pixel in the image. A hypothesis is made for a vehicle at each such candidate center, and the edge map is examined. The quality of the match between the edge map and the candidate vehicle outline is judged using a threshold referred to as the overlap threshold. If the degree of overlap exceeds the threshold, the presence of a vehicle is declared. This is followed by a suite of filters to check consistency. Thus, the overall vehicle detection process involves the specification of four parameters—the *Canny mask size, low threshold, high threshold*, and *overlap threshold*—and the model vehicle.

Formation Detection

The purpose of this module [3] is to detect vehicle formations in images. The formation detection process uses spectral analysis; spectral compliance windows are inferred from model information to search for impulsive components representing periodic object configurations such as convoys on roads or vehicles in parking lots. We have designed a detection rule on the observation space $\mathcal{O}$ which consists of the absolute spectrum magnitude associated with the impulsive component and its value relative to the median spectrum magnitude. The rule tests the dominant spectral component within a compliance window at the base and corresponding harmonic frequencies and takes into account the normalized spectrum magnitude K_a associated with the maximum peak at f^* within a compliance window and the ratio of the spectrum magnitude to the median K_{med} of this magnitude computed over the compliance window, denoted by K_r. Thus, the parameters to be optimized are K_a and K_r.

3 Detection and Training

On-line parameter training and off-line parameter optimization are set in a hypothesis-testing framework. Let H_0 and H_1 correspond to the two hypotheses (absent/present),

$$H_i : P(\mathbf{Y}|i) = P_i(\mathbf{Y})$$

The acceptance and rejection regions are designed over some observation space $\mathcal{O}$. The observation vector $\mathbf{Y}$ in the case of the vehicle detector algorithm is simply the overlap value. In the case of the formation detection algorithm, it is composed of the two spectral measures described in Section 2.

A decision rule d is simply given by $d(\mathbf{Y}) = \mathcal{I}_{\mathcal{R}}(\mathbf{Y})$, with $\mathcal{I}_{\mathcal{R}}$ the indicator function on the acceptance region $\mathcal{R}$. The acceptance region admits a parametric

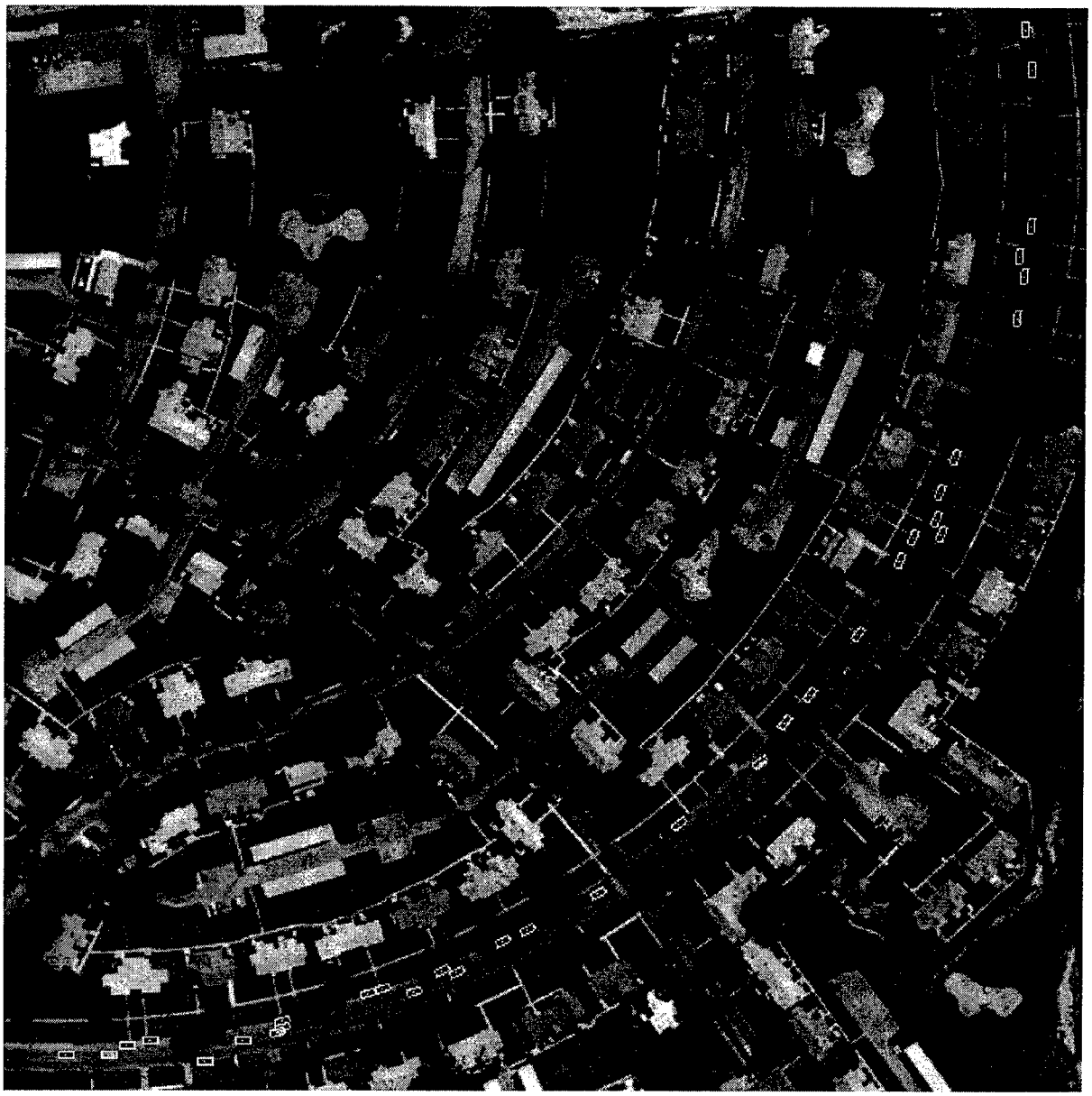


Figure 2: Vehicle detection on a road.

form

$$\mathcal{R} = \mathcal{R}_{\mathcal{V}} = \{\mathbf{Y} \text{ such that } b(\mathbf{Y}; \mathcal{V}) \geq 0\}$$

where the form of the critical/acceptance regions can be inferred from the distributions satisfied by the observation vector under either hypothesis. The design of the detection rule is set up according to the following two strategies:

- (A) Bayesian strategy:
  Find $\mathcal{V}^* = \text{argmin}(E\{R(d)\})$. This strategy consists of minimizing the expected risk $\text{argmin}(E\{R(d)\})$ [9] where the cost factors C_{fa} and C_{nd} are chosen to balance the cost associated with a false alarm and a non-detection:

  $$E\{R(d)\} = C_{nd}P_0(\mathcal{R}_{\mathcal{V}})\pi_0 + C_{fa}P_1(\mathcal{R}_{\mathcal{V}}{}^c)\pi_1$$

  where $P_0(R)$ and $P_1(\mathcal{R}_{\mathcal{V}}{}^c)$ are the false alarm and non-detection probabilities, and π_i are the priors.

- (B) Neyman-Pearson strategy:
  Find $\mathcal{V}^* = \text{argmax}(P_1(R))$ subject to $P_0(R) < \alpha$.

4 Vehicle Detector Optimization

Off-line Parameter Training
The relative impact of the parameters involved was

studied by carrying out experiments on images with known ground truths. Specifically, the empirical expected risk $E\{R(d)\}$ including both false alarm and non-detection rates was computed for the Canny mask size, the Canny thresholds and the overlap threshold. The results for the Canny mask size, the two Canny thresholds and the overlap threshold are shown in Figures 3, 4 and 5 respectively.

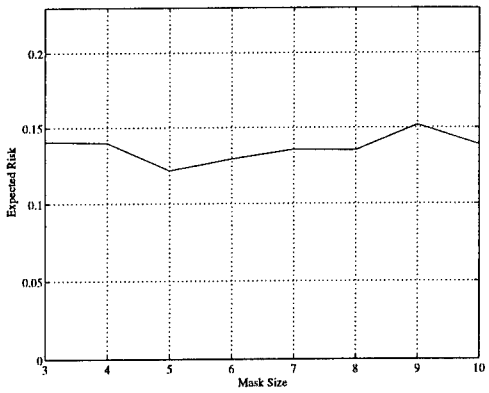


Figure 3: Vehicle detection: expected risk as a function of Canny mask size.

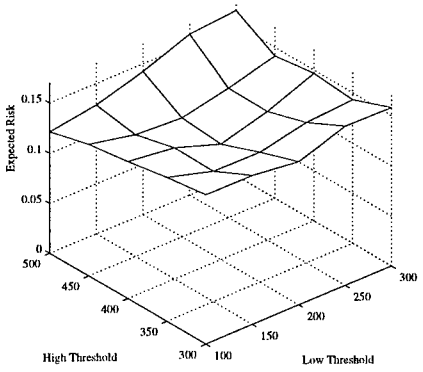


Figure 4: Vehicle detection: expected risk as a function of Canny thresholds evaluated over the Ft. Hood and Denver training sets.

From this empirical risk it was inferred that the variation in detection performance was smaller for the Canny parameters when they varied within their operational limits, while the performance varied significantly with the overlap threshold. Thus, for the former parameters it is sufficient to derive optimal estimates (off-line optimization), while for the latter, we need to consider a training tool (on-line optimization). For off-line Canny parameter optimization we use strategy (A) with the expected risk computed over the training set. We choose $C_{nd} = 0.5$ and $C_{fa} = 0.5$ and we assume equal priors, which

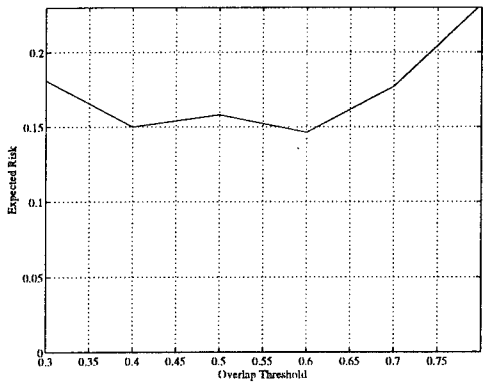


Figure 5: Vehicle detection: Expected risk as a function of overlap threshold.

is equivalent to a minimum probability error rule. The empirical value of $E\{R(d)\}$ is computed over the range of Canny parameters as a function of the Canny thresholds and mask size. The Canny parameters are set off-line to the values minimizing the expected risk. These values were found to be 5 for the mask size and $(200, 400)$ for the minimum and maximum thresholds.

On-line Parameter Calibration

For the on-line overlap threshold training, we use strategies derived from (B). We include site-specific features called "control patches" in the model of each site, which are used for automatic calibration. Empirical detection and false alarm probabilities are derived from these predetermined control patches. For each newly acquired image, the overlap threshold is automatically computed from the empirical detection and false alarm probabilities derived from the control patches as described later. Denote by $P_1(\mathcal{R}_\mathcal{V}) = r(P_1(\mathcal{R}_\mathcal{V}{}^c))$ the empirical ROC curve derived by varying the overlap threshold $\mathcal{V}$; then we want to find $\mathcal{V}^*$ satisfying $\mathcal{V}^* = \text{argmax}(P_1(\mathcal{R}_\mathcal{V}))$ subject to

- $P_0(\mathcal{R}_\mathcal{V}) < \alpha$
- $\frac{d(r(p))}{dp} \mid_{p=P_1(\mathcal{R}_\mathcal{V}{}^c)} > \beta$

The threshold is initialized to a high value (typically 0.9) and progressively reduced (typically in steps of 0.1) until one or both of the above constraints are violated. The first condition ensures that the false alarm rate will be bounded, and the second condition ensures that the slope of the ROC curve for the given value of $\mathcal{V}$ is not too small, *i.e.* that an increase in false alarm probability can be traded for a significant increase in detection probability, maximizing detection and thus establishing an upper bound on

the threshold. When a new image is acquired, the overlap threshold is automatically "calibrated" using control patches as explained above.

One issue remains to be addressed for implementation, namely, selection of control patches. In site-model-based image exploitation of aerial imagery, control patches with their associated ground truth can be specified once by representing them as specific features of the site (in terms of position, orientation, etc.). This way, whenever a new aerial image of the site needs to be subjected to any of the above algorithms, the control patches will be mapped to portions of the image, as part of the registration process (where the image is registered with the site model). The above is possible when we can identify control patches in the parking lot that are known to be always empty (such as passageways, parking lot exits, etc.), or always full (areas located near entrances). Identifying the empty control patches is easier than identifying patches with a fixed number of vehicles in them. This makes it easier to modify the empty control patch calibration module to work in batch mode. However, in this case, we would be solving a slightly different problem, because in the case of the empty patch, there exists no concept of an ROC curve as there is no P_1. In this case, the problem becomes that of finding

$$\mathcal{V}^* = \min(\mathcal{V}) \text{ subject to } P_0(\mathcal{R}_\mathcal{V}) < \alpha'$$

In our case we use $\alpha' = 0$. The threshold is initialized at a low value (typically 0.10) and progressively increased (typically in steps of 0.10) until there is no false alarm. Thus, this is a way of estimating a lower bound on the optimal threshold. This could prove very useful in cases where there is a lot of clutter in the image. The lower bound effectively cuts down the false alarm rate.

Figure 6: Batch Optimization. First Image.

Figures 6 and 7 show such a case where the learning module has been used to compute, in batch, the optimal threshold for detection in two different images (note the illumination variation), but over the same parking lot. The shaded region represents an "empty control patch" for which the ground truth is simply the fact that there are no vehicles inside the region. The control patch, being a feature set associated with the site model, is independent of the image and after the image is registered, it gets mapped to the image coordinate system. The non-shaded outline represents the actual parking lot. The results were obtained for two different images after the overlap threshold was bounded from below by means of the control patch. In both cases, the empty patch calibration overlap threshold was found to be 0.40. These results correspond to a $[P_1, P_0]$ of [0.86, 0.00] and [0.92, 0.00] respectively when run with the calibrated threshold. Consider also the case where it is not possible to identify non-empty patches with known ground truths. In this case, an interactive method could be used, where the image analyst inspects and validates the control patches over a subset of the images to be processed before initiating a batch procedure. Figure 8 shows such a case, where the image analyst delineates two patches to estimate the optimal overlap threshold to be used in future runs. The original problem was solved on the non-empty patch, for $\alpha = 0.10$ and $\beta = 0.40$, and the modified problem was solved for the empty patch

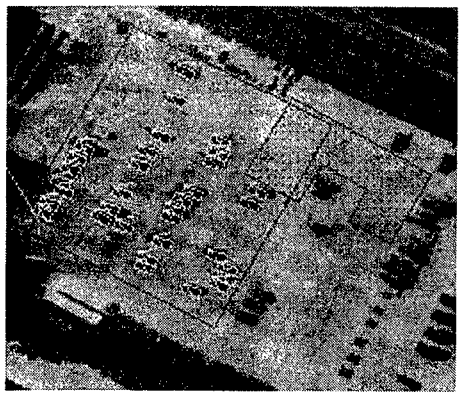


Figure 7: Batch Optimization. Second Image.

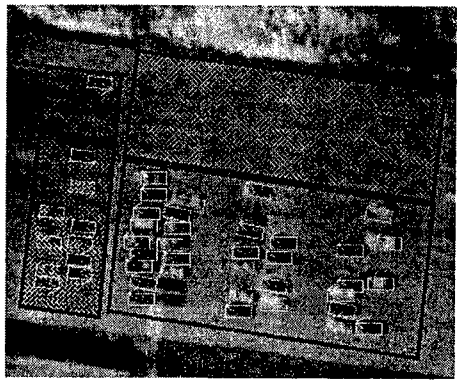


Figure 8: Non-Batch Optimization.

with $\alpha' = 0.00$. In this particular example, the empty control patch established a lower bound of 0.40 while the non-empty control patch established an upper bound of 0.4. A compatible value of 0.4 was thus used for the final run on the actual parking lot. The $[P_1, P_0]$ values were found to be [0.85,0.03].

5 Convoy Detector Optimization

In the case of the convoy detector, the components of $\mathbf{Y}$, the 2D observation vector, are the logarithms of the parameters K_a and K_r that were described in Section 2, *i.e.*$\mathbf{Y} = (\ln(K_a), \ln(K_r)) = (L_a, L_r)$. Let H_0 and H_1 correspond to the two hypotheses, with H_0 the hypothesis that no peak is present; the decision rule is simply $d(L_a, L_r) = \mathcal{I}_{R_\mathcal{V}}(L_a, L_r)$. We use a Bayesian strategy (A) for deriving the acceptance region from a training set of images. The acceptance region boundary is parameterized by vector $\mathcal{V}$ and chosen as

$$R = R_\mathcal{V} = \{(L_a, L_r), \text{such that } b(L_a, L_r; \mathcal{V}) \geq 0\}$$

Assume that the joint conditional probability distributions on $\mathbf{Y} = (L_a, L_r)$ are Gaussian, *i.e.* $H_i : P(\mathbf{Y}|i) \sim N(\mathbf{m}_i, \mathbf{\Sigma}_i), i = 0, 1$; then the log-likelihood ratio function is a quadratic function in $\mathbf{Y}$ [6], *i.e.* $(\mathbf{Y}-\mathbf{m}_1)^t\Sigma_1^{-1}(\mathbf{Y}-\mathbf{m}_1)-(\mathbf{Y}-\mathbf{m}_0)^t\Sigma_0^{-1}(\mathbf{Y}-\mathbf{m}_0)$. We assume dissimilar covariances for which the boundary equation $b(.,.;) = 0$ is a conic section. The acceptance region is determined by finding $\mathcal{V}^*$ which minimizes the expected value of the conditional risk computed over the training set, *i.e.* $\mathcal{V}^* = \text{argmin}(E\{R_\mathcal{V}(d)\})$. As an example, ten images from a particular site were chosen as a training set, and $b(.,.;)$ was assumed to be an elliptic boundary. The parameters of this elliptic boundary were optimized on the set of control images. The expected value $E\{R(d)\}$, computed over the training set, is a noisy function of V, in part due to the modest size of the training set. V* is determined by using the Nelder-Mead Simplex algorithm [8]. This function is non-convex, and therefore the simplex algorithm is not guaranteed to converge. Furthermore, the minimum is not unique. The resulting boundary for $C_{nd} = 0.55$ and $C_{fa} = 0.45$ is shown in Figure 9.

In this example, the compound detection performance yields a false alarm probability of 0.11 with a non-detection probability of 0.08.

6 Model Parameter Misspecification

We also characterized the sensitivity of parking lot occupancy detection to misspecification of the model parameters. The 3D dimensions of the vehicle were varied, and the detection and false alarm probabilities were computed on the set of test images. The resulting probabilities are displayed as functions of these dimensions in Figure 10 for the detection of active parking lots. In Figure 10, the upper surface represents the probability of detection as a function of the 3D width W and length L. The lower surface represents the false alarm probability. Situations where the width is greater than the length constitute a misspecification by $\pi/2$ of the actual vehicle orientation. In this figure we see that the resulting performance is not too sensitive to reasonable variations in size.

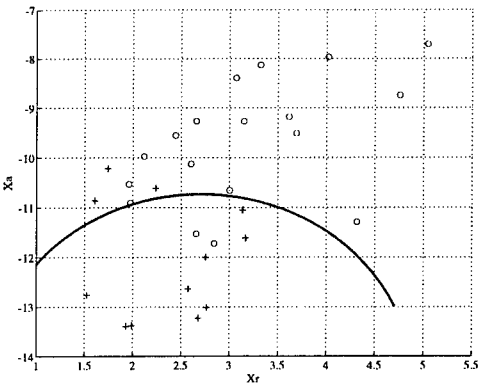


Figure 9: Decision region obtained from training images. Active parking lots are indicated by 'o's and inactive by '+'s in the (L_r, L_a) plane.

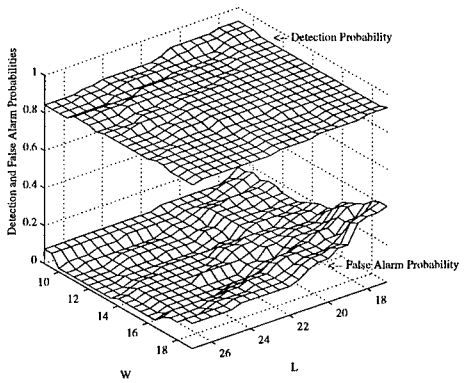


Figure 10: Sensitivity of the detection of active parking lots to misspecification of vehicle dimensions.

The situation is different for convoy detection, as seen from the simulation results in Figure 11. In this case, as the values deviate from their optimal specifications (the middle of the grid), performance degrades. As W and L increase, the false alarm probability decreases along with the detection probability. This highlights the importance of context as well as the adequate specification of model parameters for this particular application.

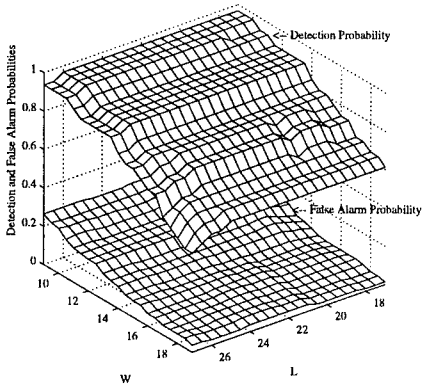


Figure 11: Sensitivity of convoy detection to misspecification of vehicle dimensions.

7 Conclusion

We have addressed the problem of automatic tuning of parameters for image understanding algorithms with specific reference to local vehicle detection and global vehicle configuration detection algorithms in AIU tasks. We proposed different optimality criteria and optimization approaches, depending upon the relative impact of the parameters on the performance of the detection algorithms, the relative impact being inferred from runs on a set of test images. Naturally, it should be ensured that the number of test images is large enough to be statistically comprehensive. This would further justify freezing parameters of lesser significance to their off-line best values. Also, in the present scheme, the sensitivity to tuning and model parameters is evaluated empirically. This study can be complemented by a more in-depth analytical study of these algorithms' sensitivity, as is done in [7] where each step of the algorithm can be described analytically and a first-order sensitivity analysis can be carried out. In the present study, empirical distributions are used. Instead, the moments can be estimated and simple hypothesis-testing techniques can be used to verify the consistency of the observed data with the assumed distribution and estimated moments. Finally, on-line parameter optimization techniques have been introduced by use of the notion of *control patches*, which is very useful and practicable in the context of site-model-based image exploitation and merits further investigation.

References

[1] P. Burlina, R. Chellappa, C. Lin, and X. Zhang, "Context-Based Exploitation of Aerial Imagery," in *Proc. Workshop on Model-Based Vision* (Boston, MA), June 1995.

[2] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 679–698, 1986.

[3] R. Chellappa, X. Zhang, P. Burlina, C. Lin, Q. Zheng, L. S. Davis, and A. Rosenfeld, "An Integrated System for Site Model Supported Change Detection," in *Proc. DARPA Image Understanding Workshop* (Palm Springs, CA), pp. 275–304, 1996.

[4] R. Chellappa, Q. Zheng, P. Burlina, C. Shekhar, and K. Eom, "On the Positioning of Multisensor Imagery for Exploitation and Target Recognition," *Proceedings of the IEEE*, Vol. 85, pp. 120–138, 1997.

[5] R. Chellappa, Q. Zheng, L. Davis, C. Lin, X. Zhang, C. Rodriguez, A. Rosenfeld, and T. Moore, "Site Model Based Monitoring of Aerial Images," in *Proc. DARPA Image Understanding Workshop* (Monterey, CA), pp. 295–318, 1994.

[6] K. Fukunaga, *Statistical Pattern Recognition*, Academic Press, 1990.

[7] X. Liu, T. Kanungo, and R. Haralick, "Statistical Validation of Computer Vision Software," in *Proc. DARPA Image Understanding Workshop* (Palm Springs, CA), pp. 1533–1540, 1996.

[8] J. Nelder and R. A. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, Vol. 7, 1965.

[9] V. Poor, *An Introduction to Signal Detection and Estimation*, Springer-Verlag, 1988.

Using RADIUS Site Models without the RCDE*

Aaron J. Heller, Christopher I. Connolly, and Yvan G. Leclerc
Artificial Intelligence Center, SRI International
333 Ravenswood Ave., Menlo Park, CA 94025 USA
E-MAIL: {heller,connolly,leclerc}@ai.sri.com

Abstract

We describe a system for converting RADIUS site models into a Web-accessible form. Once converted, the site models can be downloaded and viewed with standard, off-the-shelf Web browsers. Site models themselves are represented in the Virtual Reality Modeling Language (VRML). Each feature in the site model is cross-linked to a Web page that describes that feature in more detail, including image chips and collateral information. A demonstration of work described in this paper is accessible via the URL `http://www.ai.sri.com/~connolly/pathfinder`.

1 Introduction

The explosive growth of the World Wide Web (WWW) in just three years has transformed it into a widely accessible medium for disseminating documents, sound, images, and recently, three-dimensional models. This growth has also led to the development of a wide variety of tools for viewing and manipulating Web-accessible information. As a result, the Web is an attractive means of providing site model information.

This paper describes a system for converting RADIUS site models into a Web-accessible form. After conversion, site models can be downloaded and viewed with standard, off-the-shelf Web browsers. Site models themselves are represented in the Virtual Reality Modeling Language (VRML). Each cultural feature in the site model is cross-linked to a Web page that describes that feature in more detail, including image chips and collateral information.

1.1 RADIUS Site Models

As part of the RADIUS program, SRI has developed and assembled a suite of manual and semi-automatic tools for site-model construction that work within the RADIUS Common Development Environment (RCDE) [Heller and Quam, 1997, Heller *et al.*, 1996]. Manual techniques are those in which the 3-D model of a feature is projected into one or more images and the operator adjusts the model to align it with was is seen in the images. Semi-automatic techniques are those in which an operator provides an initial rough estimate of a feature's position, size and topology and the system then refines or extends the model of the object using information extracted from the image(s).

Figure 1 shows a portion of a typical RADIUS site model. The majority of the features modeled fall under three broad categories:

- Buildings and other structures such as bridges, and petroleum and water storage tanks.
- Lines of Communication such as roads, railroad tracks, and other linear features such as rivers and streams.
- Functional Areas such as parking lots, site

*This work was sponsored by SRI International. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the United States Government, or SRI International.

perimeters, rail transfer points and other area features such as forested areas.

Figure 2 shows a typical construction sequence for a building using the Model-Based Optimization system [Fua, 1996]. Similar techniques are available for extraction of linear and area features.

1.2 The Virtual Reality Modeling Language

1.3 What is VRML?

The VRML Frequently Asked Questions (FAQ) [1] list says:

> VRML stands for Virtual Reality Modeling Language. It is usually pronounced "V-R-M-L," but its friends pronounce it "vermel." The goal of VRML is to create the infrastructure and conventions of cyberspace, a multi user space of many virtual worlds on the Net.
>
> VRML 1.0 is a subset of the Inventor File Format (ASCII) with some additions to allow linking out to the Web and including other URLs. The linking out feature (WWWAnchor) provides the same feature that HREF anchors provide in HTML. Another critical feature was the LOD (level of detail) which allows the right amount of data for an object based on how prominent it is in the scene, or the rendering speed of the browsing machine.

Our current interest in VRML is a bit more pragmatic. While it is clearly not a replacement for richer scene description data models, such as the Synthetic Environment Data Representation and Interchange Specification (SEDRIS) [The SEDRIS Team, 1996] now being developed by the Defense Modeling and Simulation Office (DMSO), it is a simple and low-cost method to export geometric site-model data from the RCDE in a form that is suitable for dissemination via the WWW and usable with a wide variety of freely-available browsers and rendering tools.

[1] `http://vag.vrml.org/VRML_FAQ.html`

1.3.1 History

The development of VRML grew out of a discussion among several attendees of the first annual World Wide Web Conference in 1994. VRML was seen as a three-dimensional extension of HTML. As with most Web-related projects, the development of a VRML specification and prototype was rapid. A draft specification was released in fall of 1994. The format chosen for VRML was Silicon Graphics' ASCII *Open Inventor* format. A specification for VRML 1.0 was finalized and released on May 26, 1995 [Pesce, 1995]. Not surprisingly, most of the initial work on VRML took place at Silicon Graphics. Over the next two to three years, between 20 and 30 VRML 1.0 browsers were developed for exchanging geometry over the Web.

1.3.2 Current status

By August of 1996, a specification for VRML 2.0 was released [Bell *et al.*, 1996]. While VRML 1.0 is capable of representing static geometry, VRML 2.0 was aimed at augmenting geometry with time-varying behavior. Because of its increased complexity, only 4 browsers are known to exist for VRML 2.0, as of this writing.

The structure of a VRML scene description consists of an identifier string, to distinguish VRML 1.0 from VRML 2.0 descriptions, followed by a sequence of Node specifications. Each node represents a geometric feature or an attribute of that feature. For example, there are Cube, Sphere, Cylinder, and Cone nodes for representing those shapes. In addition, point sets can be specified. Faces (IndexedFaceSet nodes) are described as lists of points that form polygons. Material properties, certain transformations, viewpoints, and lighting properties can also be specified.

1.3.3 Strong Points

An important advantage to using VRML 1.0 for disseminating site models is that many VRML 1.0 browsers have been developed across a wide variety of platforms. Although the VRML 1.0 specification has minor ambiguities, browsers exhibit generally uniform behavior, that is, VRML models will have the same or similar appearance across browsers. As

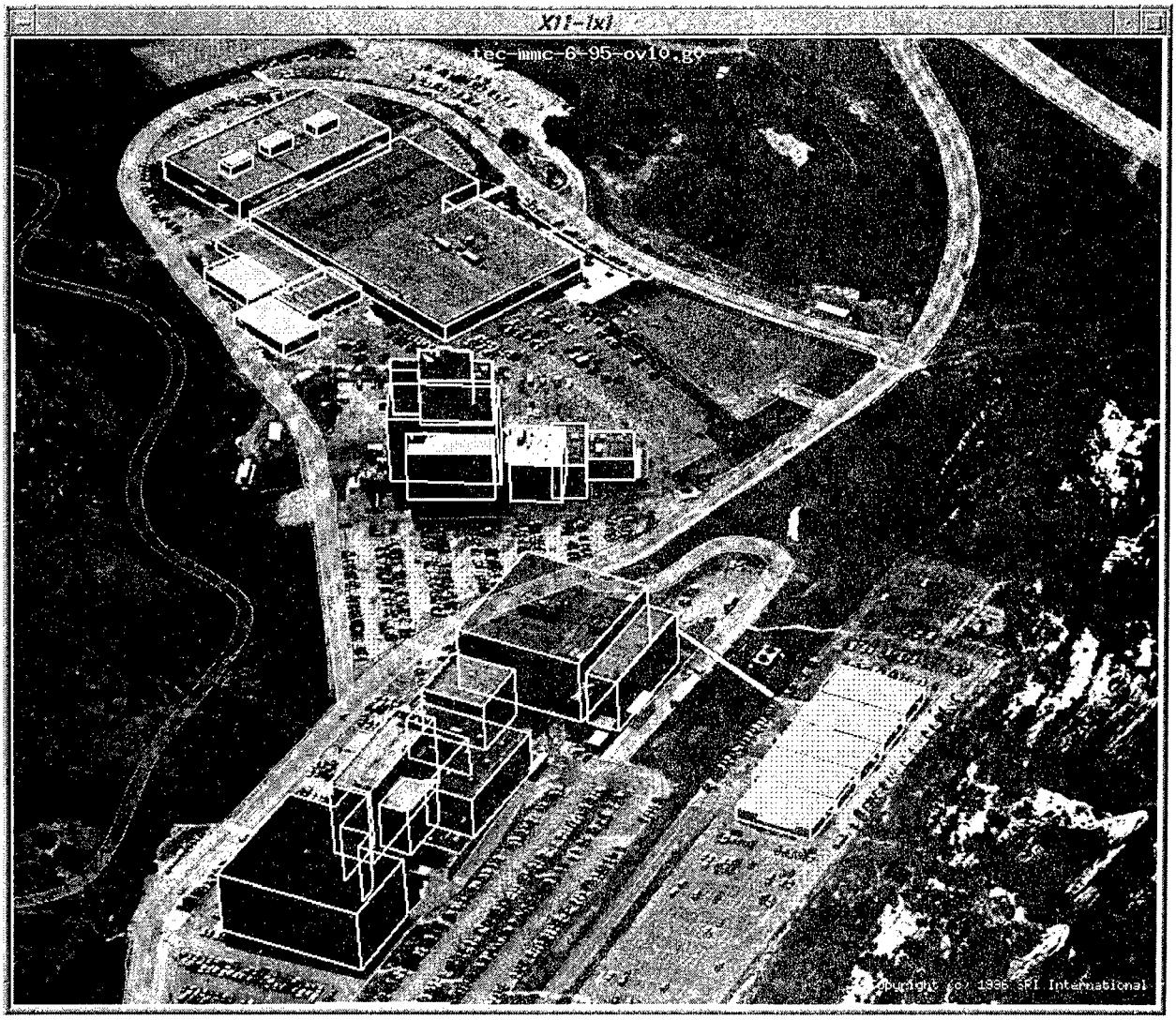


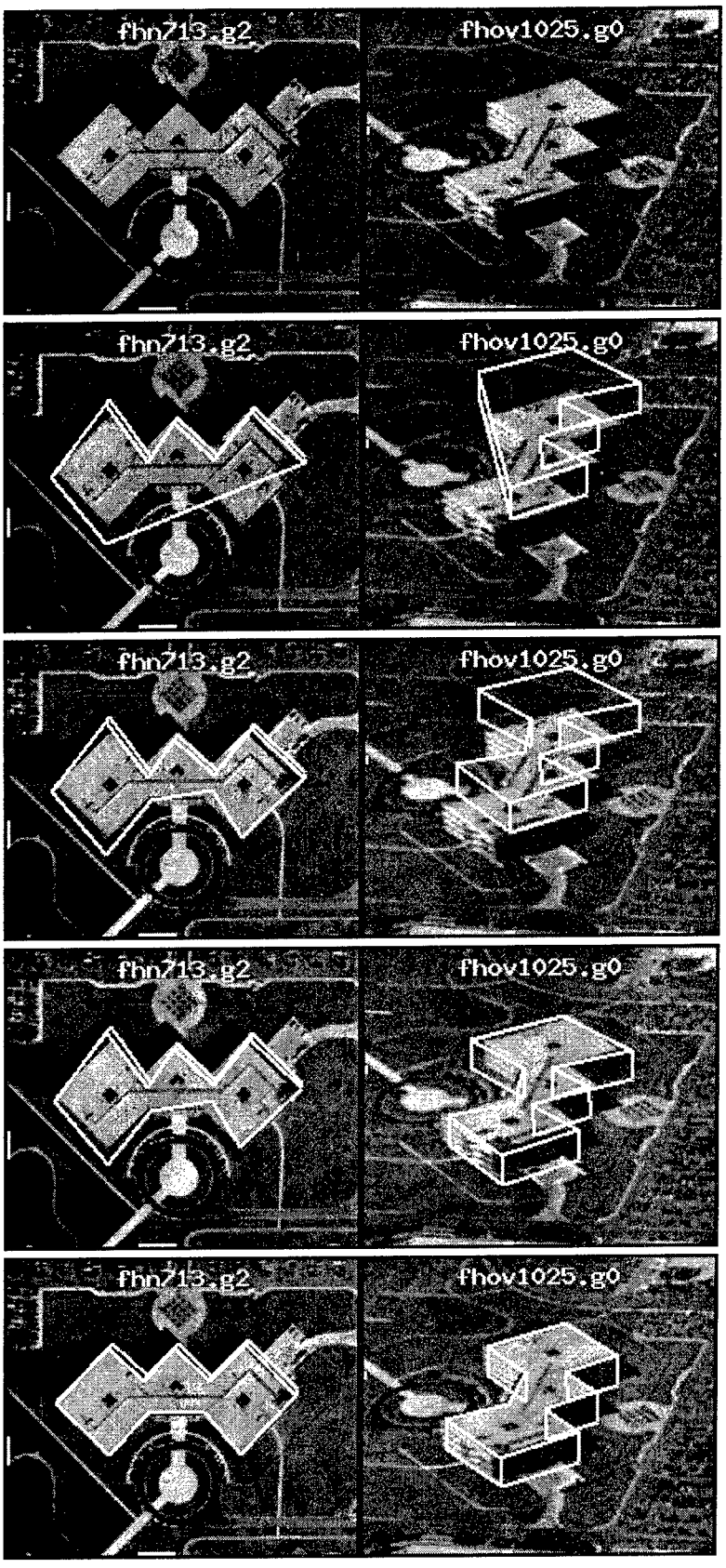
Figure 1: A portion of a typical RADIUS site model, projected onto an image of the site. The site model mainly consists of buildings, linear and area features.

a result, VRML 1.0 has become a satisfactory, if imperfect, standard for exchanging geometry via the Web. VRML 1.0 also allows an association between objects and Web URLs. As with hypertext Web pages, mouse clicks on an object can be used to retrieve and display Web pages that are associated with the selected object.

1.3.4 Weak Points

The main deficiency of VRML for our application is that there is no provision for georeferencing the data in the models. The VRML spec includes an Info node, which is intend to be used for comments. We press this into service to store the site-model's *LVCS-to-geocentric-transform*, to allow the recovery of geopositioning information and translation into standard cartographic coordinate systems.

Unfortunately, VRML 2.0 removes some features that were found to be useful in VRML 1.0. Transformations have been restricted rather than generalized: 4×4-matrix transformations are permitted in VRML 1.0, but not in VRML 2.0. Moreover, in VRML 2.0, transformations cannot be sequentially specified and composed; they must be nested recursively, resulting in a bulkier representation for site models. As of this writing, there are no VRML 2.0 browsers that implement the complete specification.



The original images.

Sketch roof-line *(Add Vertex)*.

Done with roof-line *(Drop)*.

Correct elevation *(MBO–Z-Search)*.

Optimize Shape *(MBO–Opt)*... Done!

Figure 2: The sequence of steps used to model a complex-shaped building with the extrusion primitive and the SRI-authored Model-Based Optimization system. This entire sequence typically takes less than one minute of elapsed time.

This is in part due to the introduction of behaviors and scripting into VRML 2.0, which greatly complicates the semantics of the language. The VRML 2.0 standards process also appears to be dominated by commercial interests. As a result, VRML 2.0 is targeted mainly toward low-bandwidth home users browsing the Internet. The scientific visualization community appears to have had little influence in specifying the VRML 2.0 standard.

2 WWW Site Visualization Tool

2.1 The Translator

The translator is implemented in approximately 3000 lines of Common Lisp that runs in the RCDE and operates on site-models that have been loaded into the system. Besides making the code somewhat simpler, this allows us to translate models regardless of whether they were loaded from RCDE feature-set files, the RADIUS Testbed System database, or created in the current session. The translation process in implemented as a single pass over the objects in the site-model. Only those objects which are "present" in the selected view are translated. This allows the RCDE's feature set mechanism and associated menus to be used to select the set of objects to be translated.

Every object in an RCDE site-model has a *object-to-world-transform* which is used to transform the coordinates of the object's vertices to the site's *local-vertical-coordinate-system* (LVCS). This transform is translated into a VRML transform node. One complication is that VRML uses a right-hand coordinate system in which the *y-axis* is up whereas the convention used by RCDE is that *z-axis* is up with the *x-axis* pointing east.

In addition, the any face of an RCDE object may have a texture map associated with it. The translator automatically creates the texture images and calculates the texture coordinates for inclusion in the VRML node describing the object. These texture maps can be either "inlined" in the VRML file or can be written into separate files that are referenced via a URL from the VRML file.

3-D objects that have a direct representation in VRML, such as cubes, cylinders, and spheres are simply translated into the corresponding node type. Other, more complicated objects, such as houses and complex buildings, are represented in the RCDE with a face-edge-vertex (f-e-v) datastructure, called the *planar-solid* class. This class of object is specialized by introducing parameterized constraints among the vertices and faces. The parameterized representation is used for adjusting the object and the f-e-v representation is use for drawing the object. Because of the availability of the f-e-v representation a single method suffices to translate all classes of compact 3-D objects into VRML indexed-face-sets.

Roads and fences are represented by ribbon-curve objects in RCDE, which are comprised of a sequence of vertices that trace the centerline of the object and a width (or height for fences) at each vertex. VRML does not have a corresponding node-type, so ribbons are triangulated and then translated into VRML indexed-face-sets.

The RCDE represents terrain as regular quad-mesh or tri-mesh objects. Since the faces in a quad-mesh object are not necessarily planar, we use tri-mesh object for terrain and translate these to indexed-face-sets.

If a *sun-direction* vector is present on the selected view, it is translated into a VRML DirectionalLight node and added to the scene file.

2.2 HTML Generation

For each feature in the VRML representation, a URL is created containing an HTML page with collateral information for that feature. This page also contains image chips displaying the feature as it appears in all available images associated with the site. Image chips are generated by first collecting those site images within which the feature is visible. The extent of the feature in each image can then be determined by using the feature's bounding box, and transforming this (via the world-to-image transformation) into the corresponding image coordinates. The chip is created by windowing into the corresponding image, and converting this window into a GIF file for use in the HTML page. The feature attributes can be used to populate this page with appropriate text. The URL containing this page is then attached to the VRML representation by creating a WWWAnchor node linking the feature to its own

Web page.

3 Example

Figures 3 through 5 show an example result. The initial page introduces the tool and lists the available site-models (Fig. 3a). The user has selected the "Lockheed-Martin Corp., Denver" site-model and is shown a page containing a synoptic view of the site and several alternative versions of the models (Fig. 3b). These have been generated to accommodate different network bandwidths and browser capabilities, ranging from small models that contain only the modeled objects, to fully texture-mapped models with high-resolution DTED. After selecting the version "with texture and terrain," the VRML browser (in this case SGI's WebSpace) is started and the 3-D model is displayed (Fig. 4). At this point the user can "fly around" and inspect the model from any viewpoint. As the mouse is placed on the individual objects, they are highlighted and their names appears in a information window at the bottom of the browser. Clicking on any of the objects, causes the HTML browser to retrieve the page of attributes, metadata, and image chips for the selected object (Figs. 5 a&b). The "Modifications" and "Comments" fields on these pages, allow the user to enter data which is sent back to the server and ultimately incorporated into the site-model.

4 Related Work

TerraVision II is the primary application of the recently started MAGIC–II project. It is an extended version of the TerraVision terrain visualization application [2] that was created for the original MAGIC project[3].

TerraVision was designed to visualize a single rectangular geographic area represented by elevation and ortho–rectified image data at multiple levels of detail. These data, typically larger than 1 G–Byte in size, are distributed across a high–speed network. TerraVision accesses the data in real time from the network as the user moves across the terrain, thus giving the illusion that all of the data is stored locally. This approach allows users access to very large amounts of data without the need to copy, store, and maintain the data locally.

TerraVision was inflexible because it was difficult to create a dataset of a very large area where different areas would have data at different resolutions. It was also impossible to use multiple types of images that could be fused together under user control.

To overcome this inflexibility, TerraVision II is designed to use composite datasets consisting of many image pyramids, each of which can be created and stored independently at different sites. The "glue" that holds them together are VRML files. In a sense, TerraVision II will be an enhanced VRML browser that will be able to handle very large, network–based, multi-resolution datasets. However, it will have additional capabilities for merging different image types (under user control) and it will produce seamless renderings of scenes with multiple levels of detail. We expect that an important source of data for TerraVision II will be RADIUS site-models produced by the VRML production system described here.

5 Conclusions

Even though we have struggled with a number of deficiencies, ambiguities, and errors in the VRML specification and browsers, and are, in general, unhappy with the standards process and the direction in which VRML is evolving, we have still found it and the freely-avaiable browsers to be a useful mechanism for visualizing RADIUS site-models and illustrating possible methods for disseminating geospatial information on the WWW. Readers are encouraged to form their own opinions by experimenting with the example discussed in this paper. It is accessible via the URL `http://www.ai.sri.com/~connolly/pathfinder`.

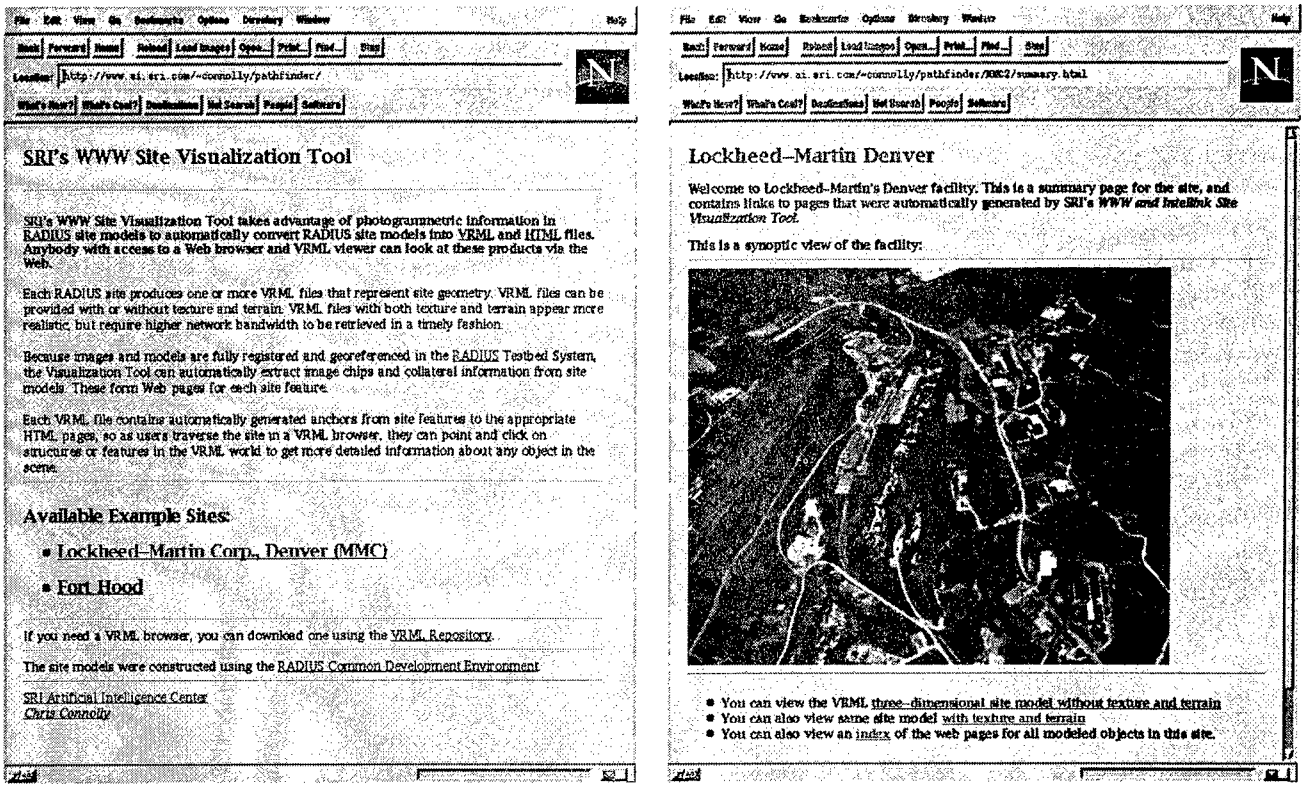
References

[Bell *et al.*, 1996] Garvin Bell, Rikk Carrey, and Chris Marrin. The virtual reality modeling language specification version 2.0. URL: `http://vag.vrml.org/VRML2.0/FINAL/`, 1996.

[Fua, 1996] P. Fua. Cartographic Applications of Model-Based Optimization. In *DARPA Image Understanding Workshop*, Palm Springs, CA,

[2] `http://www.ai.sri.com/~magic/terravision.html`

[3] `http://www.magic.net/`



(a) (b)

Figure 3: The homepage for the WWW Site Visualization Tool and initial page for the Lockheed-Martin Denver site. The URL is `http://www.ai.sri.com/~connolly/pathfinder`

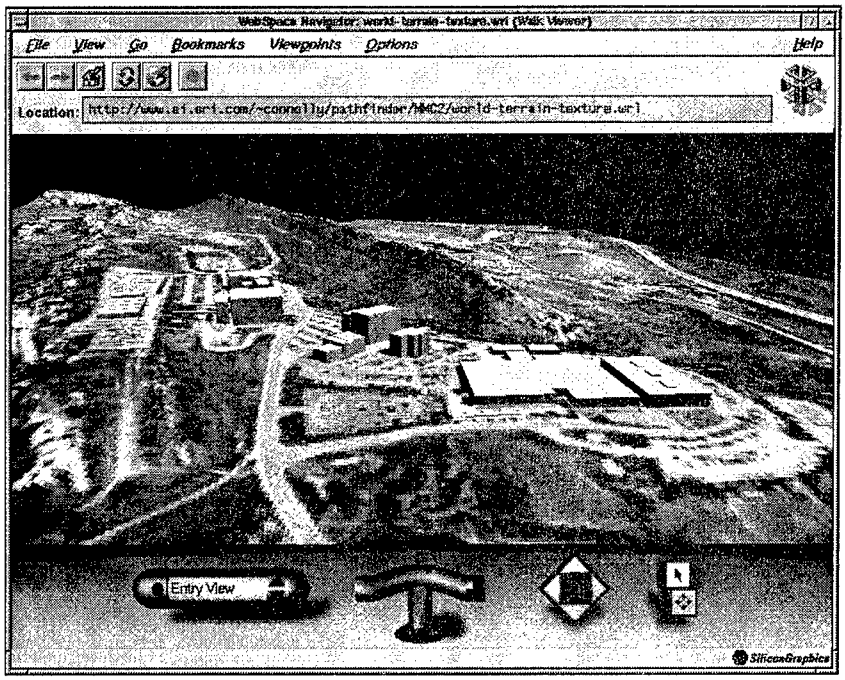
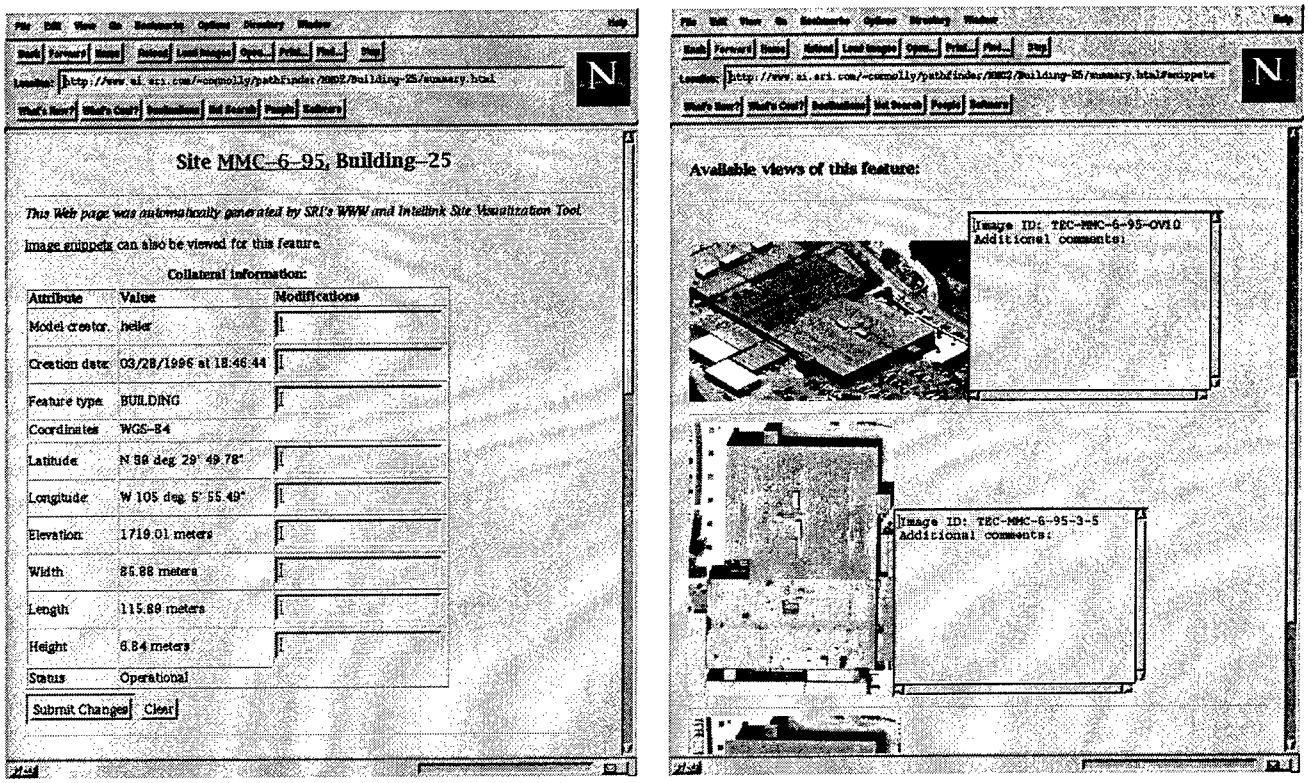


Figure 4: A texture-mapped, 3-D rendering of the site-model created from the VRML description of the site and displayed by SGI's WebSpace.



(a) (b)

Figure 5: The a sample of the Web pages that are hyperlinked to every cultural feature in the site model. The text and image chips are generated automatically from the attributes, metadata, and images stored with the site-model in the RCDE.

February 1996. Morgan Kaufmann.

[Heller and Quam, 1997] Aaron J. Heller and Lynn H. Quam. The RADIUS Common Development Environment. In Oscar Firschein and Tom Strat, editors, *RADIUS: Image Understanding for Imagery Intelligence*. Morgan Kaufmann, San Mateo (CA), 1997.

[Heller *et al.*, 1996] A. J. Heller, P. Fua, C. Connolly, and J. Sargent. The Site-Model Construction Component of the RADIUS Testbed System. In *DARPA Image Understanding Workshop*, pages 345–355, 1996.

[Pesce, 1995] Mark Pesce. *VRML: Browsing and Building Cyperspace*, chapter Appendix A: VRML: The Virtual Reality Modeling Language Version 1.0 Specification. New Riders, Indianapolis, 1995. Also available at the URL: `http://vag.vrml.org/vrml10c.html`.

[The SEDRIS Team, 1996] The SEDRIS Team. Synthetic environment data representation and interchange specification (sedris). URL: `http://www.sedris.net/`, 1996.

Grouping Planar Projective Symmetries

R.W. Curwen, J.L. Mundy *
G.E. Corporate Research and Development
1 Research Circle
Niskayuna, NY 12309

Abstract

A novel approach to grouping symmetrical planar curves under a projective transform is described. Symmetric curves are important as a generic model for object recognition. Often it is desirable to represent an object by a generic description rather than an accurate geometric model, for example when describing a general class of objects by the properties of the class. Symmetry is a well defined generic representation. It is intuitively easy to understand, is recurrent in real world objects, and provides strong image constraints. We describe a geometric construction which reduces the complexity of recovering a projective transform between images of non-trivial planar curve sections when correspondence for two lines is given. This enables detection and grouping of planar symmetrical curves in perspective images.

1 Symmetry as a Generic Model

Generic models are vital to the future of object recognition. Techniques which rely on accurate, three dimensional, geometric models of the objects to be recognized are difficult to adapt to the rapid pace of change in the real world. Often such models are not available, or are uneconomical to produce, but the imagery consumer would still like to be shown all images containing "an aircraft". Symmetry is a well defined, intuitively accessible generic model. It is pervasive in imagery because a symmetrical object is both statically and dynamically more stable. Statically stable objects such as chairs and tables are good examples, and many structures, such as arches and domes, exhibit strong symmetry. Dynamically stable objects such as aircraft, cars, birds, etc, are also symmetrical. It is also economical to repeat structures in order to minimize the number of required component designs and manufactured parts, leading to symmetry in images. Thus it is efficient to represent objects by their symmetries, and the detection of these symmetries in an image is an important step towards generic object recognition.

An object recognition system must group image features according to the underlying scene structure in order to recognize them as an object. Symmetry is a key scene structure, and is reflected in the scene projection. The detection of a symmetry is a strong cue to feature grouping, either within a single object, or across a spatial group of related objects. Figure 1 shows several images of objects and scenes which exhibit symmetry. Often several forms of symmetry are found in a single image, and if the context of the scene is sufficiently constrained *a priori*, then these symmetries may be enough to support reconstruction. For example, Ulupinar and Nevatia [Ulupinar and Nevatia,

*This work was supported by DARPA contract F33615-94-C-1021, monitored by Wright Patterson Airforce Base, Dayton, OH. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the United States Government, or General Electric.

1992] define three kinds of 3-D symmetry: parallel symmetry; line-convergent symmetry and skew symmetry. They use the constraints from these symmetries for 3-D reconstruction from 2-D curves. Even if insufficient constraint is available to allow reconstruction, it may be possible to discriminate between a number of possible states using symmetry. An example of discrimination is continuously monitoring a region in which many types of aircraft are parked. The classes of aircraft are unknown *a priori*, but the planar bilateral symmetry of the wings on the aircraft is sufficient to determine whether the region is occupied.

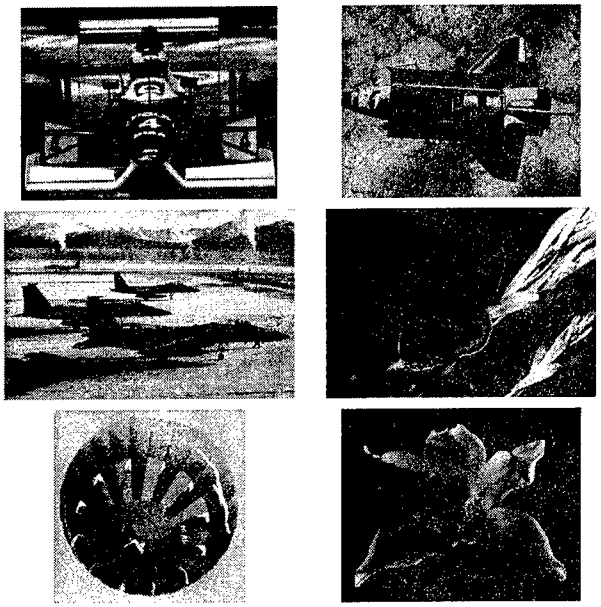


Figure 1: Examples of symmetry, sometimes approximate, seen in diverse images

Methods for determining curve correspondence have been proposed based on the Generalized Hough Transform [Ballard, 1981], which results in an intensive search through a multi-dimensional parameter space, and on differential [Cham and Cipolla, 1995b] and semi-differential invariants [Van Gool *et al.*, 1992]. Cham and Cipolla [Cham and Cipolla, 1995a] have investigated the detection of affine symmetries and the establishment of curve correspondences. In particular they show how geometric saliency of symmetries can be used to characterize the discriminatory power of pairs of symmetrically grouped curves.

2 Grouping a Pair of Planar Curves

Figure 2 shows the general planar symmetry considered in this paper. The two images C and C' of planar curves $\mathbf{C}$ and $\mathbf{C}'$ are related by a 3×3 projectivity T. Given two curves in the image, it is this projectivity T which must be recovered. The constraint that two curves are related by a projectivity is obviously only a strong constraint if the curves are sufficiently complex. For example, all pairs of conics are related by a projectivity, so all conics will appear symmetric in this manner. For curves more complex than a conic, however, a projectivity relation between curve pairs is a strong cue for grouping.

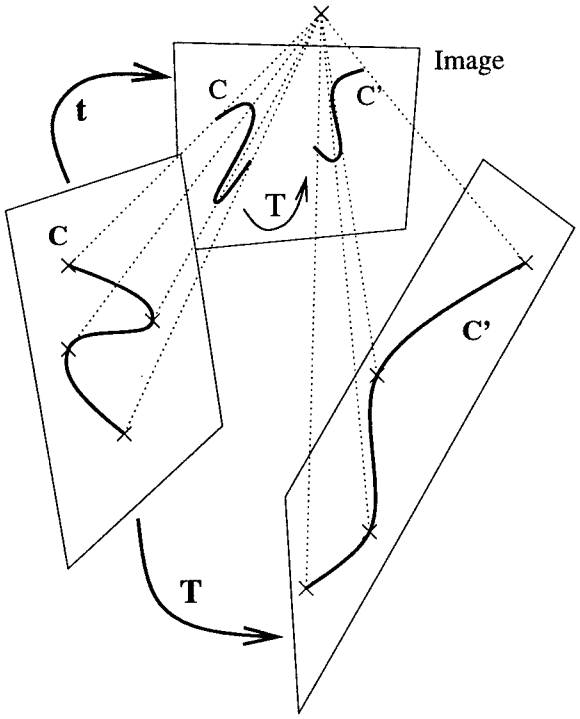


Figure 2: The planar symmetry considered for grouping. Here planar curves $\mathbf{C}$ and $\mathbf{C}'$ are related by a projection $\mathbf{T}$, and both are viewed as images C and C' under projection $\mathbf{t}$. The transform T from C to C' is a 3×3 projectivity.

2.1 The Correspondence Problem

One of the central problems in curve grouping is the correspondence problem. Curves C and

C' are not recovered from the image with the same parameterization, so the point on C' corresponding to a given point on C is unknown. Certain special points on curves are known to be invariant under projection, such as points of inflection and bitangent lines. Thus if a point on C is an inflection, then it must correspond to an inflection on C'. But the correspondence other than between such special points is not known. For example, if a pair of curves is recovered from an image, the ends of one curve might be occluded, so it is certainly not the case that the ends of each curve correspond. If the curves are lines, this is the well known aperture problem. Under a Euclidean transform for any curve other than a line or a circle, the correspondences may be recovered by finding the unique transform. Similarly under a projective transform, the correspondences of a pair of curves more complex than conic should be recoverable.

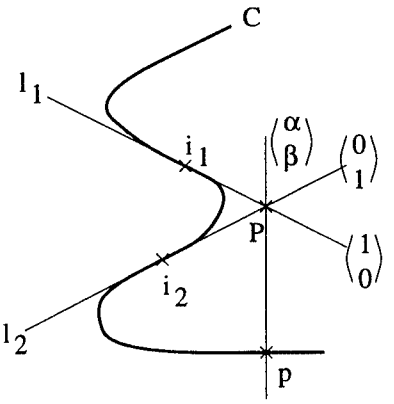


Figure 3: A pencil of lines defined by the tangents at two inflection points, l_1 and l_2, to the curve C. The coordinates of a line within this pencil, $q = (\alpha, \beta)$, are defined in the basis defined by the two tangent lines. Thus any point p on the curve C is assigned a coordinate q based on the line in the pencil passing through p.

2.2 Rational Parameterization

Consider two inflection points i_1 and i_2 on C. Construct the tangent lines l_1 and l_2 at these points. Assuming these lines are distinct, they define a basis for a pencil of lines P passing through their intersection, and any line l in P may be written as a linear combination of the two basis lines:

$$l = \alpha l_1 + \beta l_2 = \begin{bmatrix} l_{11} & l_{21} \\ l_{12} & l_{22} \\ l_{13} & l_{23} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

The basis lines have coordinates $(1,0)$ and $(0,1)$ respectively within this pencil. Now take any other point p on the curve C, and construct the line in P which passes through p. This line, and therefore the point p, will have a coordinate $q = (\alpha, \beta)$ in the pencil P. Writing $l_1 = (l_{11} l_{12} l_{13})^t)$ and $l_2 = (l_{21} l_{22} l_{23})^t)$ we have

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} l_1^t p \\ l_2^t p \end{bmatrix} \quad (1)$$

Since the lines are expressed in homogeneous coordinates, $q = (\alpha, \beta)$ are also homogenous, so any multiple of q will be the same line in the pencil. We will refer to this as the *pencil space* $P(l_1, l_2)$. The construction of this pencil is illustrated in Figure 3.

Now consider the ratio α/β. The construction described assigns a value α/β to each point on the curve. This is not necessarily a parameterization: if multiple points on C lie on the same line of the pencil then they will share the same α/β. A curve is of genus zero if a rational parameterization of the curve exists. For example, all conics may be rationally parameterized, and so are of genus zero. However, for the purpose of this study it is not necessary for the curve to be parameterized by α/β.

2.3 Matching Curves in Pencil Space

Let us return the to example of two curves, C and C' related by a perspectivity T. For an arbitrary parameterization of C, point $p(s)$ on C corresponds with point $p'(s)$ on C', and using homogeneous coordinates we have

$$p'(s) = \lambda_p T p(s).$$

Here λ_p represents the scaling ambiguity of homogeneous coordinates. The inflection point tangents on C will also be transformed by the same projectivity:

$$\begin{aligned} l_1' &= \lambda_1 T^{-t} l_1 \\ l_2' &= \lambda_2 T^{-t} l_2 \end{aligned}$$

where T^{-t} is used to denote the transpose of the inverse of T. Thus if the correspondence of the inflection points is assumed, we can set up a pencil space for each curve $P(l_1, l_2)$ and $P(l_1', l_2')$. Now $p(s)$ and $p'(s)$ have coordinates $q(s) = (\alpha(s), \beta(s))$ and $q'(s) = (\alpha'(s), \beta'(s))$, given by equation 1:

$$\begin{bmatrix} \alpha(s) \\ \beta(s) \end{bmatrix} = \begin{bmatrix} l_1^t p(s) \\ l_2^t p(s) \end{bmatrix}$$
$$\begin{bmatrix} \alpha'(s) \\ \beta'(s) \end{bmatrix} = \begin{bmatrix} l_1'^t p'(s) \\ l_2'^t p'(s) \end{bmatrix}$$

We can solve these equations to give the relation between the curves in the pencil spaces.

$$\begin{bmatrix} \alpha'(s) \\ \beta'(s) \end{bmatrix} = \begin{bmatrix} \lambda_1\lambda_p & 0 \\ 0 & \lambda_2\lambda_p \end{bmatrix} \begin{bmatrix} \alpha(s) \\ \beta(s) \end{bmatrix} \quad (2)$$

So the pencil space curves $q(s)$ and $q'(s)$ are related by an anisotropic scaling. Indeed, since the pencil spaces are homogeneous, there is only one degree of freedom in equation 2. This is far simpler to solve than the original, eight degree of freedom problem. Figure 4 illustrates this matching process for a curve which has been scaled and reflected. This result is closely related to the semi-differential invariants proposed by Van Gool et al. [Van Gool *et al.*, 1992, Van Gool *et al.*, 1991] in which a projectively invariant parameterization of curves, known as *Arc Length Space* is constructed using a mixture of reference points and curve derivatives.

2.4 Computing Correspondence from Pencil Space

In the pencil space representation, all that is required is to find a scaling λ such that $(\lambda\alpha(s), \beta(s))$ is a uniform scaling of $(\alpha'(s), \beta'(s))$. Points which are in correspondence can then be found by taking the intersection of both curves with rays through the origin. However, these rays will in general intersect each curve multiple times, unless the curve is genus zero. These multiple intersections with the curve are equivalent to multiple points on the curve with the same rational parameterization. Furthermore only part of one curve may be visible, or the curves may be occluded. Thus the calculation of λ should be robust to extra or missing curve portions.

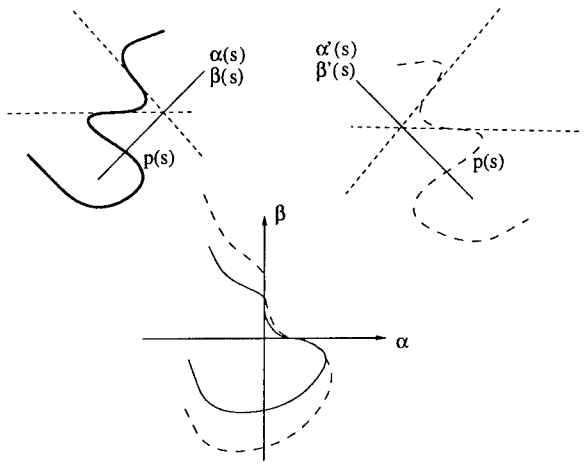


Figure 4: For two curves with a pair of corresponding inflection points idendified, matching proceeds by construction of the pencil space representation of each curve, in which the two curves will be related by an anisotropic scaling.

One approach to the problem is to match the curves using a Hausdorff distance [Huttenlocher *et al.*, 1993], solving for both scaling factors rather than just a single one. This will guarantee the correct assignment of correspondences over the curve pairs. Another approach is to minimize the minimum angle between tangents to the curves along rays from the origin. Specifically, for each point $q(s)$, construct the ray through the origin, and intersect that ray with the second curve, resulting in n points $q'(t_i), i = 1 \ldots n$. Compare the tangent orientation at $q(s)$ with the tangent orientation at $q'(t_i)$, and pick the closest pair. Calculate the total angular distance between tangents over all $q(s)$, and similarly the reverse distance from q' to q. Minimize this distance over λ, a multiple of one coordinate of one curve.

2.5 Verifying the Symmetry

Once the correspondences have been found, the corresponding curve points on C and C' can be used to solve for the complete perspectivity T. This can then be used, with a Borgefors

matcher [Borgefors, 1986] to find other curves which support the same symmetry transform. However, the two curves which have been corresponded might not form a well conditioned basis for the projectivity. For example, if the curves are from a bilateral symmetry and are both approximately perpendicular to the axis of symmetry, then the transform found will not be reliable except in the immediate vicinity of the original curves. Multiple corresponded curves might be combined to solve for the transform in such a case, or an iterative process might be used to incrementally include new points into the transform calculation. This problem is beyond the scope of this paper.

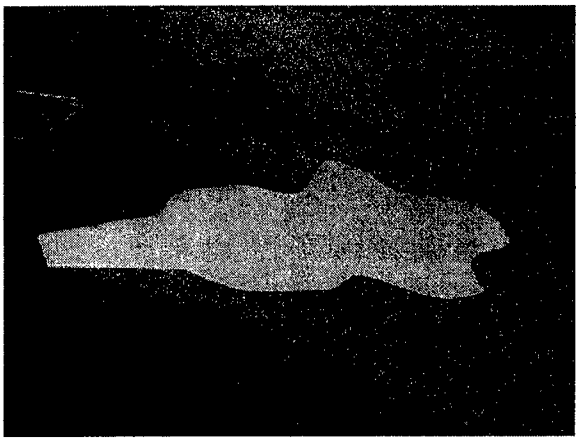
3 Experimental Results

We have tested this method of curve matching on a number of images. These images were initially segmented, and then the two contours to be matched were selected by hand. B-spline curves were fitted to the curve segments [Vanroose *et al.*, 1995, Dierckx, 1993], and these were used to detect inflection points. Two appropriate inflection points were then selected by hand on both curves. The pencil space representation of the two curves was then constructed, and the scaling factors automatically determined to match the two curves.

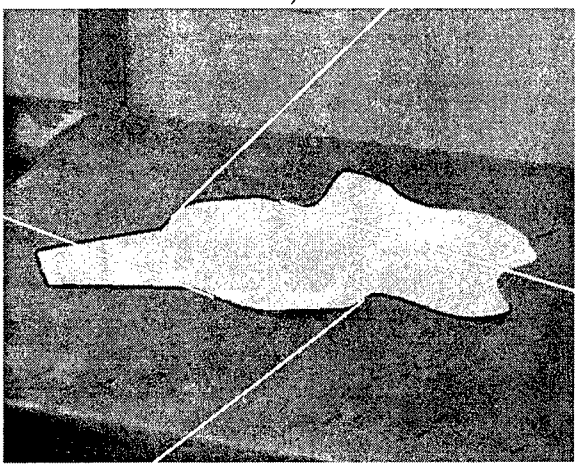
Figure 5, shows the first example, a simple planar shape, but with significant perspective distortion. The inflection points are used to construct the pencil space representations in Figure 6, which are matched by the anisotropic scaling. The correspondences thus found are shown on the original curves in Figure 7.

Whilst the location of the recovered inflection point on the curve is not robust, the tangent at the inflection point is a well known, since the curve has zero curvature at the inflection point. Thus the tangents at inflection points are a good basis for forming the pencil of lines.

However, the two line correspondences in the image pairs need not be confined to be tangents at inflection points. Many objects are composed of straight lines, and these lines can be detected in the segmentation and similarly used as the



a)



b)

Figure 5: The original image is shown in a). In b) the image has been segmented, B-splines fitted to the curves, and the inflection points and their tangents extracted. The four tangents shown in white are used to parameterize the two halves of the shape in black.

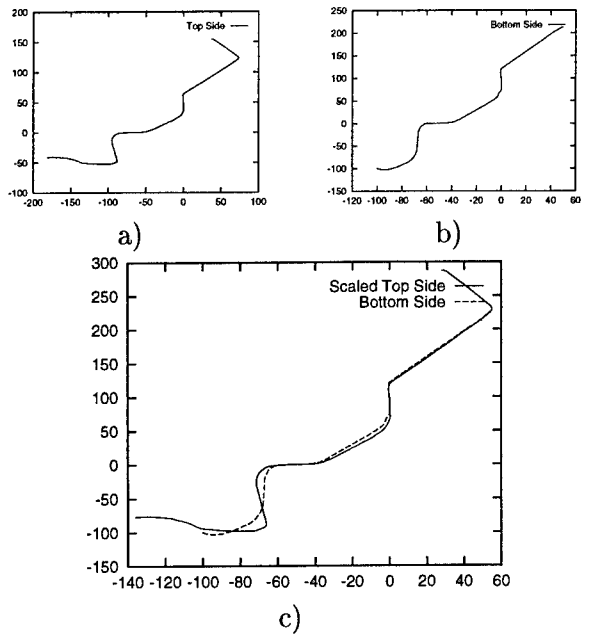


Figure 6: The graph in a) is of α versus β for the top curve. The graph in b) is of α' versus β' for the bottom curve. In c) the anisotropic scaling of (α, β) has been found to bring the two curves into alignment.

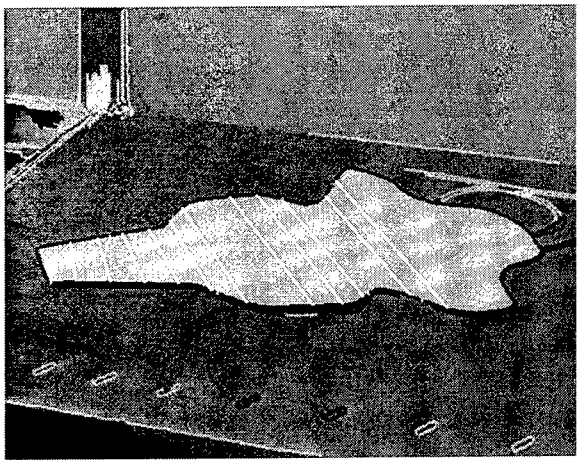
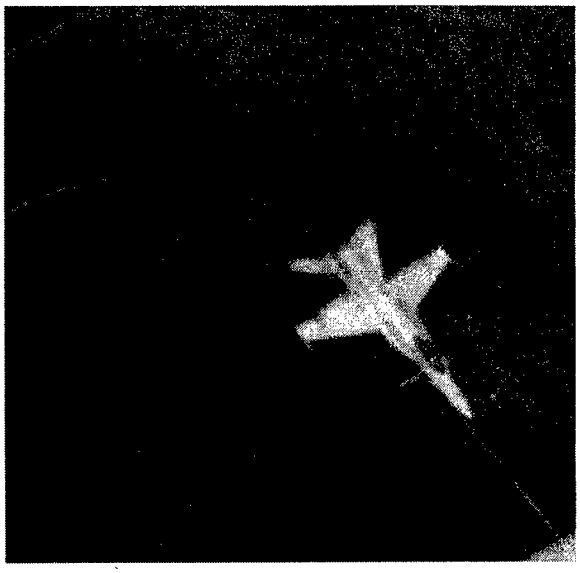


Figure 7: The correspondence found in pencil space is then used to find the perspectivity between the original image curves. Also shown in black are putative supporting edges, ie. they map onto other edges through the same transform. These would be examined for consistency and might be used to grow the symmetrical group.

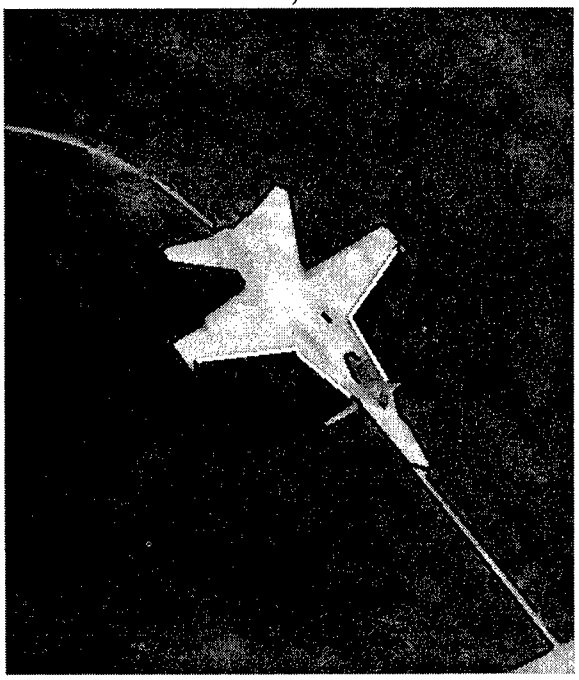
bases of the pencils. Figure 8, shows an example of the use of image lines. Here there is only moderate perspective distortion, but the outline of the aircraft is not strictly planar. The lines shown are used to construct the pencil space representations in Figure 9, which show some deformation after matching because the original curves are not planar. A good scaling is found nevertheless, and the correspondences are shown on the original curves in Figure 10.

As was discussed in Section 2.5, it is possible that the curve sections for which correspondence is recovered do not constrain the perspectivity sufficiently to give a good global transform. An example of this behaviour is shown in Figure 11. Here the front of a butterfly's wings are used in an attempt to group the complex curves of the wings into a single symmetric group. Again some inflection point correspondences are manually chosen, and used to construct the pencil space curves in Figure 12, which match rather well after scaling. The correspondences are shown on the original curves in Figure 13, but the transform is only valid for the region around those curves. Elsewhere, as is shown, the transform is far from correct. This transform is a bilateral symmetry, and thus is not well constrained by the curves which are close to colinear. To completely define the transform would require two such lines, which would meet at the vanishing point.

Finally a control result is presented in Figure 14. Here the planar shape used in Figure 5 is modified so that the only region of the shape which is symmetrical is the convexity bounded by the two inflection points used. The rest of the curve, whilst similar in shape, is no longer projectively symmetric. Figure 15 shows the pencil space curves for this case, and evidently the two curves are not related by an anisotropic scaling, except in the region of the symmetric convexity. Compare these curves to those shown in Figure 6.

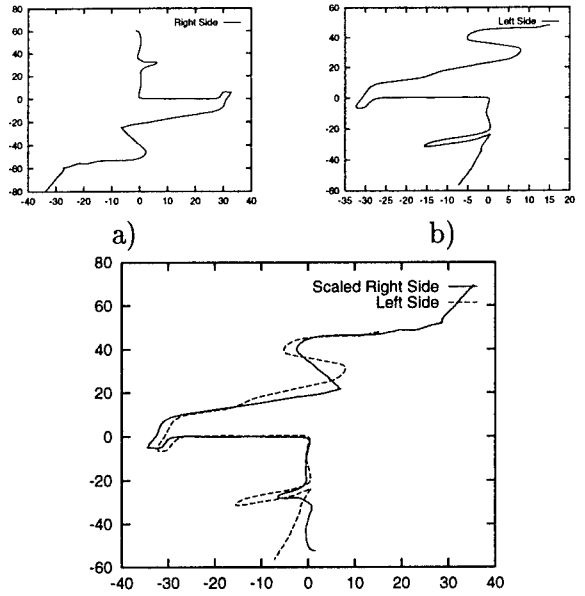


a)



b)

Figure 8: The original image is shown in a). In b) the image has been segmented, and straight lines fitted to find the four lines shown in white. These lines are used to parameterize the two halves of the aircraft in black.



a) b)

c)

Figure 9: The graph in a) is of α versus β for the right hand side of the aircraft. The graph in b) is of α' versus β' for the left hand side of the aircraft. In c) the anisotropic scaling of (α, β) has been found to bring the two curves into alignment.

4 Symmetry for Model Supported Exploitation

Symmetry is a powerful tool for site monitoring, under the model supported exploitation paradigm [Bremner *et al.*, 1996, Mundy and Vrobel, 1994, Mundy *et al.*, 1993]. Often an image analyst is unable to give an exact geometric model of the event or object for which they are searching. In such cases a more generic vocabulary for describing the event is desirable. Symmetry is one example of the ways in which an object can be described without providing a CAD model.

Consider the scenario where a new aircraft is known to be under development, and the analyst wishes to find images of the aircraft on the runway. A model of the aircraft is not known, indeed such a model may be the required product, but it is almost certain to exhibit a strong bilateral symmetry. The analyst would then place a region of interest at the end of the runway at the site, and ask to be notified

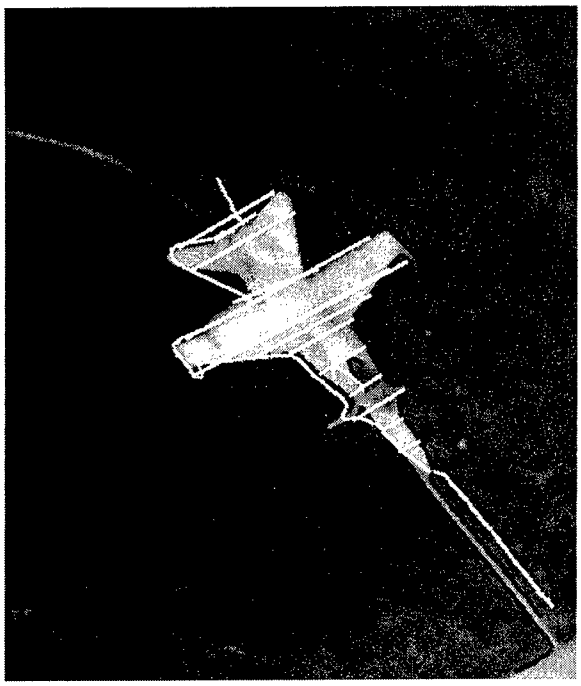
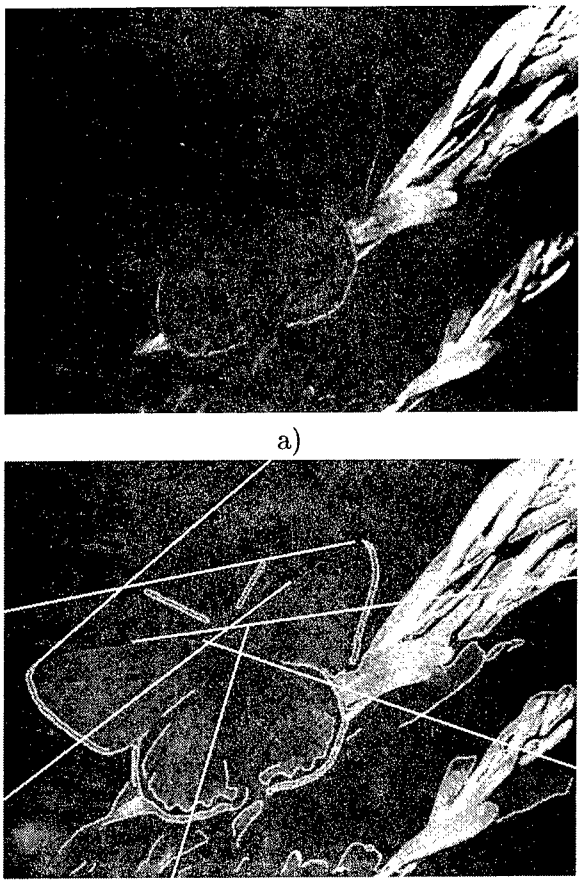


Figure 10: The correspondence found in pencil space is shown by the lines in white. Also shown in white is the right hand side curve transformed by the recovered perspectivity, which lies close to the left hand side as expected.



a)

b)

Figure 11: The original image is shown in a). In b) the image has been segmented, B-splines fitted, and inflections found for the front of the wings. The inflections used are shown in white on the black curves.

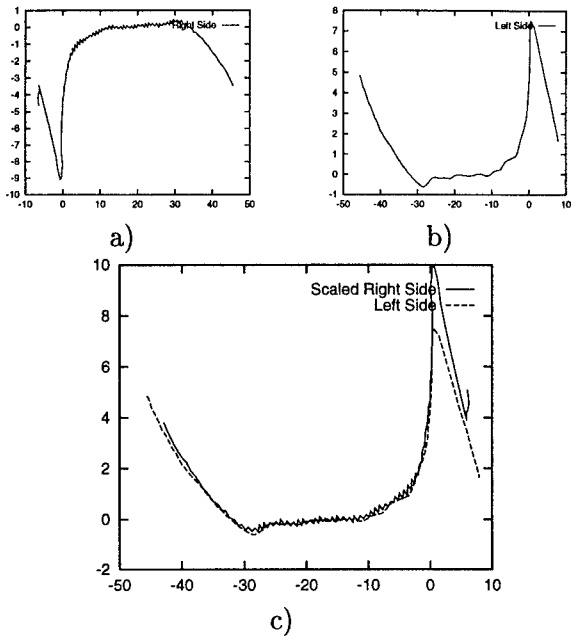


Figure 12: The graph in a) is of α versus β for the right hand curve. The graph in b) is of α' versus β' for the left hand curve. In c) the anisotropic scaling of (α, β) has been found to bring the two curves into alignment.

Figure 13: The correspondence found in pencil space is shown by the lines between the original curves in black. Also shown are some other edges from the left wing, and their projection under the recovered transform. Clearly the transform is not correct globally, though it is correct in the vicinity of the grouped curves.

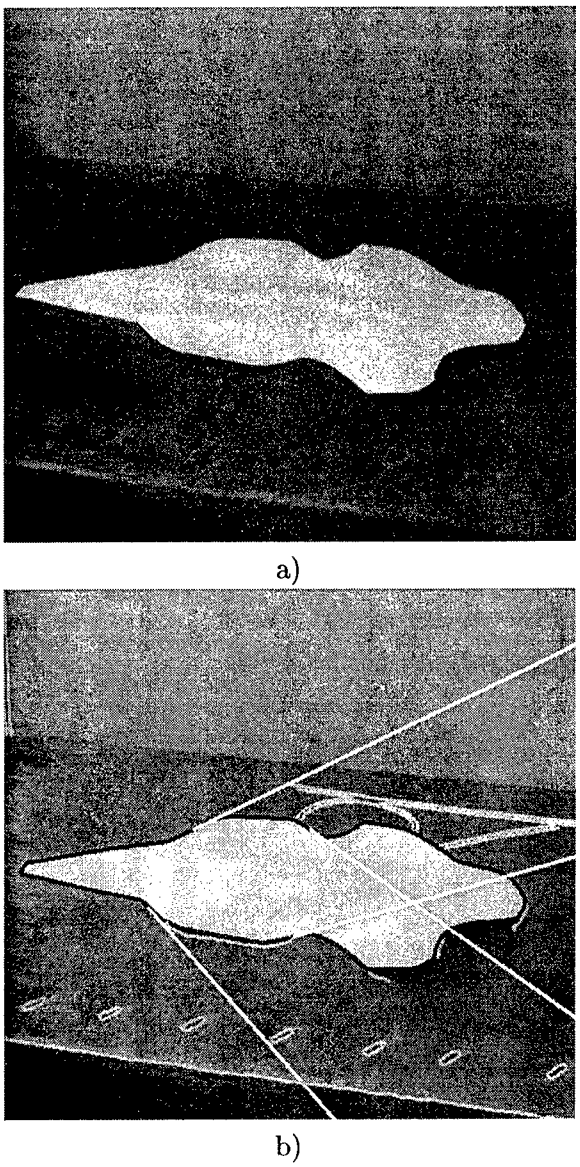


Figure 14: An asymmetric example is shown in a). In b) the image has been segmented, B-splines fitted, and inflections found for the two curves. The inflections used are shown in white on the black curves.

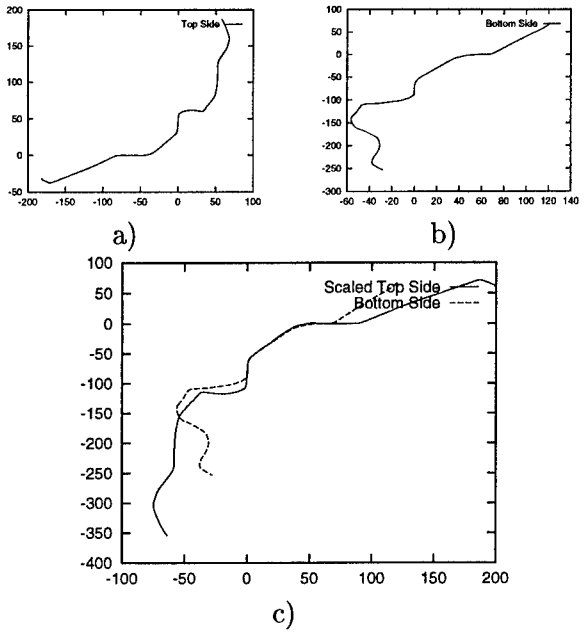


Figure 15: The graph in a) is of α versus β for the top hand curve. The graph in b) is of α' versus β' for the bottom hand curve. In c) an anisotropic scaling of (α, β) has been manually selected to bring the two symmetric portions of the curves into alignment, but the asymmetric portions are obviously not related by an anisotropic scaling.

when an image is captured in which a bilaterally symmetric object is present in the region. The strength of the bilateral symmetry can be measured by finding the percentage of edgels within the region which are explained by the symmetry.

Acknowledgments

The authors wish to thank Charles Stewart for his advice and for many inciteful conversations throughout this work, and Valery Snell for proof reading drafts of the paper.

References

[Ballard, 1981] D. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13:111–122, 1981.

[Borgefors, 1986] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34:344–371, 1986.

[Bremner *et al.*, 1996] W. Bremner, A. Hoogs, and J Mundy. Integration of image understanding exploitation algorithms in the radius testbed. In *Proc. ARPA Image Understanding Workshop*, page 255, 1996.

[Cham and Cipolla, 1995a] T-J. Cham and R. Cipolla. Geometric saliency of curve correspondences and grouping of symmetric contours. Technical Report CUED/F-INFENG/TR 235, University of Cambridge, Department of Engineering, October 1995.

[Cham and Cipolla, 1995b] T-J. Cham and R. Cipolla. Symmetry detection though local skewed symmetries. *Image and Vision Computing*, 13(5):439–450, 1995.

[Dierckx, 1993] P. Dierckx. *Curve and Surface Fitting with Splines.* Monographs on Numerical Analysis. Clarendon Press, Oxford, 1993.

[Huttenlocher *et al.*, 1993] D. P. Huttenlocher, J.J.Noh, and W. J. Rucklidge. Tracking non-rigid objects in complex scenes. In *Proc. Fourth International Conference on Computer Vision*, 1993.

[Mundy and Vrobel, 1994] J. Mundy and P. Vrobel. The role of iu technology in radius phase ii. In *Proceedings IUW*, 1994.

[Mundy *et al.*, 1993] J. Mundy, R. Welty, L. Quam, T. Strat, B. Bremner, M. Horwedel, D. Hackett, and A. Hoogs. The radius common development environment. In *ARPA Image Understanding Workshop*, 1993.

[Ulupinar and Nevatia, 1992] F. Ulupinar and R. Nevatia. Recovery of 3-d objects with multiple curved surfaces from 2-d contours. In *Proc. DARPA Image Understanding Workshop*, pages 627–633, January 1992.

[Van Gool *et al.*, 1991] L. Van Gool, P. Kempenaers, and A. Oosterlinck. Recognition and semi-differential invariants. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 454–460, Hawaii, 1991.

[Van Gool *et al.*, 1992] L. Van Gool, T. Moons, E Pauwels, and A Oosterlinck. Semi-differential invariants. In J. L. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, pages 157–192. MIT Press, 1992.

[Vanroose *et al.*, 1995] P. Vanroose, L. Van Gool, and A. Oosterlinck. A system for projectively invariant recognition of planar objects. In *Proceedings of SPIE's International Symposium on Optical Science, Engineering and Instrumentation: Vision Geometry IV*, volume 2573, pages 379–390, San Diego, July 1995.

User Interface Representations For Image Understanding

Michael A. J. Puscar and Anthony Hoogs
Building 10, Room 1954
Lockheed Martin Corporation
P.O. Box 8048
Philadelphia, PA 19101
[puscar|hoogs]@mds.lmco.com

Abstract

To enable the transition of exploitation image understanding (IU) technology into near-operational use in the intelligence community, user interfaces must be designed to allow simple, intuitive access to IU functionality and results. The complexity of IU systems, both in required inputs and processing, must be hidden from the user as much as possible to avoid heavy training costs.

This paper describes some of the important user interface issues encountered when image understanding algorithms are introduced to an imagery analyst, and discusses some of the solutions that have evolved during the development of the RADIUS Testbed. Significant issues we have encountered are algorithm and parameter selection, algorithm execution, visual representation of change, and display of historical results.

1 Introduction

Research and Development for Image Understanding Systems (RADIUS) is a research project, funded by ARPA and other government organizations, aimed at increasing the efficiency of image analysts (IA) by using IU technologies[1]. RADIUS uses site models, a set of three dimensional wire frame objects which outline features of interest in a common geographic location. Rather than executing IU algorithms on entire images, RADIUS uses site models to narrow the focus of the IU algorithms to pixels located in a specific region of interest.

The goal of the RADIUS user interface was to develop a consistent, concise, and intuitive human-computer interface (HCI), given the complexity and diversity of IU systems. This HCI design had to allow the IA to use IU exploitation algorithms without requiring knowledge of IU or the principals of individual algorithms. This goal was achieved by segmenting the image understanding architecture into two elements; a general piece that is common to all algorithms and algorithm-specific components. These two pieces are mirrored in the RADIUS user interface.

1.1 Feature Profiles

A *feature profile* can be defined as a collection of criteria used to monitor an item of intelligence interest through image understanding. Developed as part of the Exploitation IU Framework [3,4,5] for the RADIUS program, feature profiles are the cornerstone for all exploitation algorithm execution in the RADIUS testbed.

Each profile has a number of associated elements, kept general enough so that they may be used with any algorithm. These elements include a name, associated feature of interest,

such as a building, road, or parking lot, a required confidence level, algorithm sensitivity level, and a brief description.

In addition, every feature profile is associated with a specific algorithm, which is assigned by the IA at the time of creation. The RADIUS Testbed System (RTS) contains a variety of IU algorithms, each varying in performance requirements and capabilities.

As an example, an IA may wish to monitor a weapons factory for possible bomb damage. The feature profile in this scenario would contain a building/structure presence algorithm and the feature of interest (in this case, the weapons factory). Additional information could also be added to this feature profile, such as a required confidence level, sensitivity level, and a short description.

1.2 Algorithm and Parameter Selection

When the IA discovers a new structure or region of interest that he would like to monitor automatically, he must begin by creating the profile. Since profiles contain many complicated elements, some of which directly affect an IU algorithm's performance, the user interface is designed in a way that would characterize the components of the profile effectively.

Under the guidance of the National Exploitation Laboratory (NEL) HCI Document[2], the profile creation menu was designed in a top-down fashion. This was done to allow readability for an analyst who may not be familiar with the workstation. An *Instructions* line guides the user as to what the next step will be in profile creation, and there is a *Suggest* line available to give the user hints about the various applicable menu choices. Further documentation is also available when holding the menu cursor above any of the widget fields[6].

Profile creation includes individual algorithm selection. Though the IA may know the region he is interested in running IU on, he may not be familiar enough with the specific IU technologies to know which algorithm to choose. This, in particular, presents a formidable problem, since picking an IU algorithm in the wrong situation may not produce the desired result.

To make this process easier for the IA, a one line, 130 character documentation string is available for each individual algorithm. This documentation string is not meant to detail the inner workings of the IU. Instead, it provides a brief understanding of each algorithm's effectiveness. The RTS User Manual can be consulted if a more detailed explanation of the image understanding technologies are required.

One of the challenges in developing the HCI for profile creation was the ability to handle a multitude of algorithm specific components without further complicating the user interface. Rather than adding these components to the profile creation menu, a more efficient solution is to keep as many elements as possible in a common structure, then branch into algorithm specific parameter menus as necessary.

1.3 Profile Execution

The IA can execute IU on incoming imagery from the profile main menu. This menu contains all of the profiles owned by the current user, along with a status for each profile. The status of a profile determines whether it will be executed during the execution stage. Profiles with a status of *ON* are considered to be active. Profile execution consists of executing multiple profiles on multiple imagery in one session. An example of the profile main menu can be seen in Figure 1.

In designing the user interface for the profile execution stage, there are some important factors that must to be considered. First, the IA needs to have an easy way to select a group of target imagery. This imagery, however, must also be consistent with each algorithm's individual constraints.

This problem was solved by invoking an additional menu whenever a profile execution occurs. This menu allows the user to select a sub-

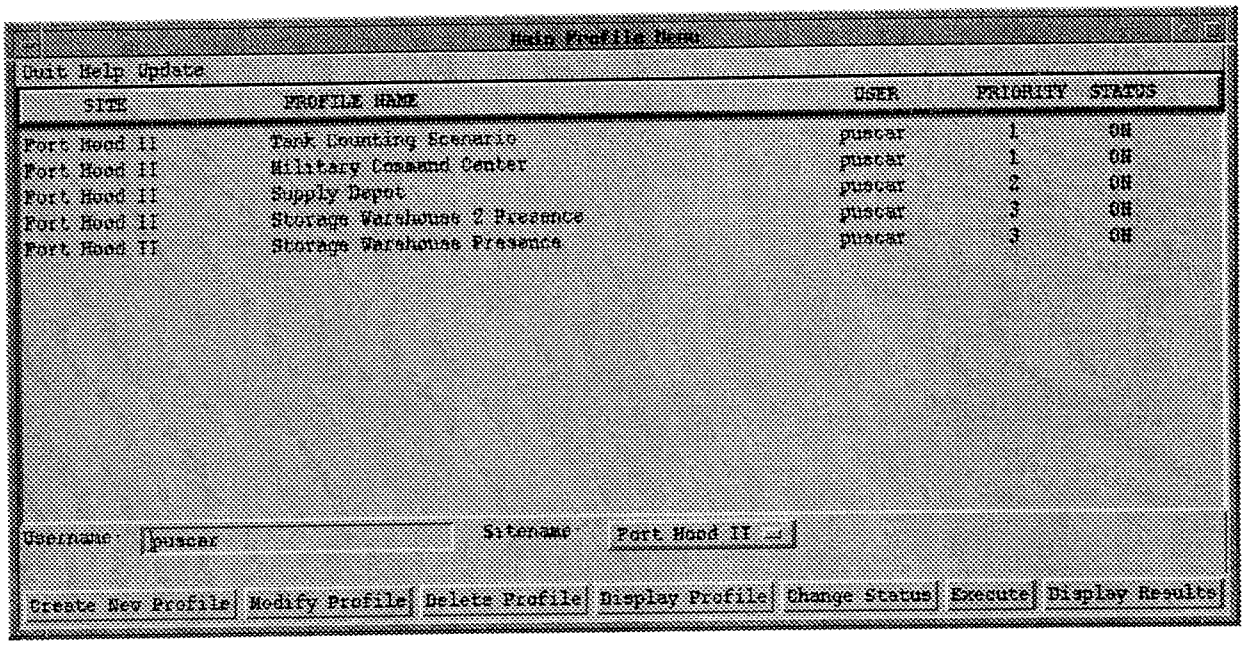


Figure 1: A sample profile main menu containing multiple profiles.

set of imagery by inputing a date range. An example of the *Additional Constraints* menu is shown in Figure 2.

When the IA enters a date range, an automatic query is sent to the RTS Database[5]. This query not only retrieves all imagery within the given date range, it also filters out imagery that does not match individual constraints of the algorithms involved in execution. One example might be an algorithm that is only effective when run on a given sensor (EO imagery, for example). Furthermore, incoming imagery may not contain the target structure or region of a profile. Images not containing the object of interest are discarded automatically.

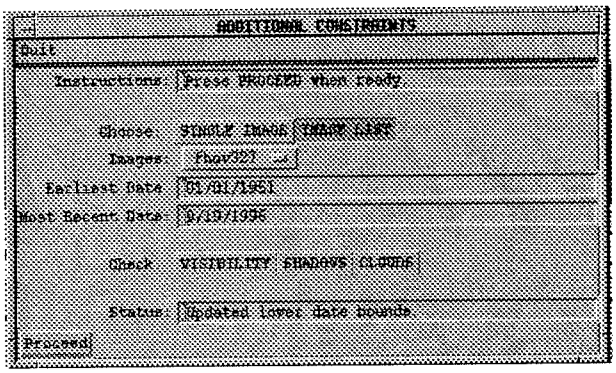


Figure 2: Sample additional constraints menu.

1.4 Conditions Affecting Algorithm Execution

In addition to algorithm constraints, there are other imagery conditions that may inhibit algorithm execution. Three main concerns were identified and accounted for during the design of the profiles system.

In order for an algorithm to execute to its fullest potential, a clear line of visibility must be available in the given image. In some cases, the viewpoint to the IA's region of interest may be obstructed by another object. Before profile execution occurs, the IA has the option of checking visibility conditions automatically, so that algorithms will only be executed on areas with a defined viewpoint of the entire object.

Even if there is a clear line of visibility to a target object, shadowing conditions may also inhibit some algorithm executions. The IA has the option to run an automatic shadow checking algorithm on each image that IU will be run on. If the object of interest is shadowed in a given image, the profile system will not execute any algorithms on that object in that particu-

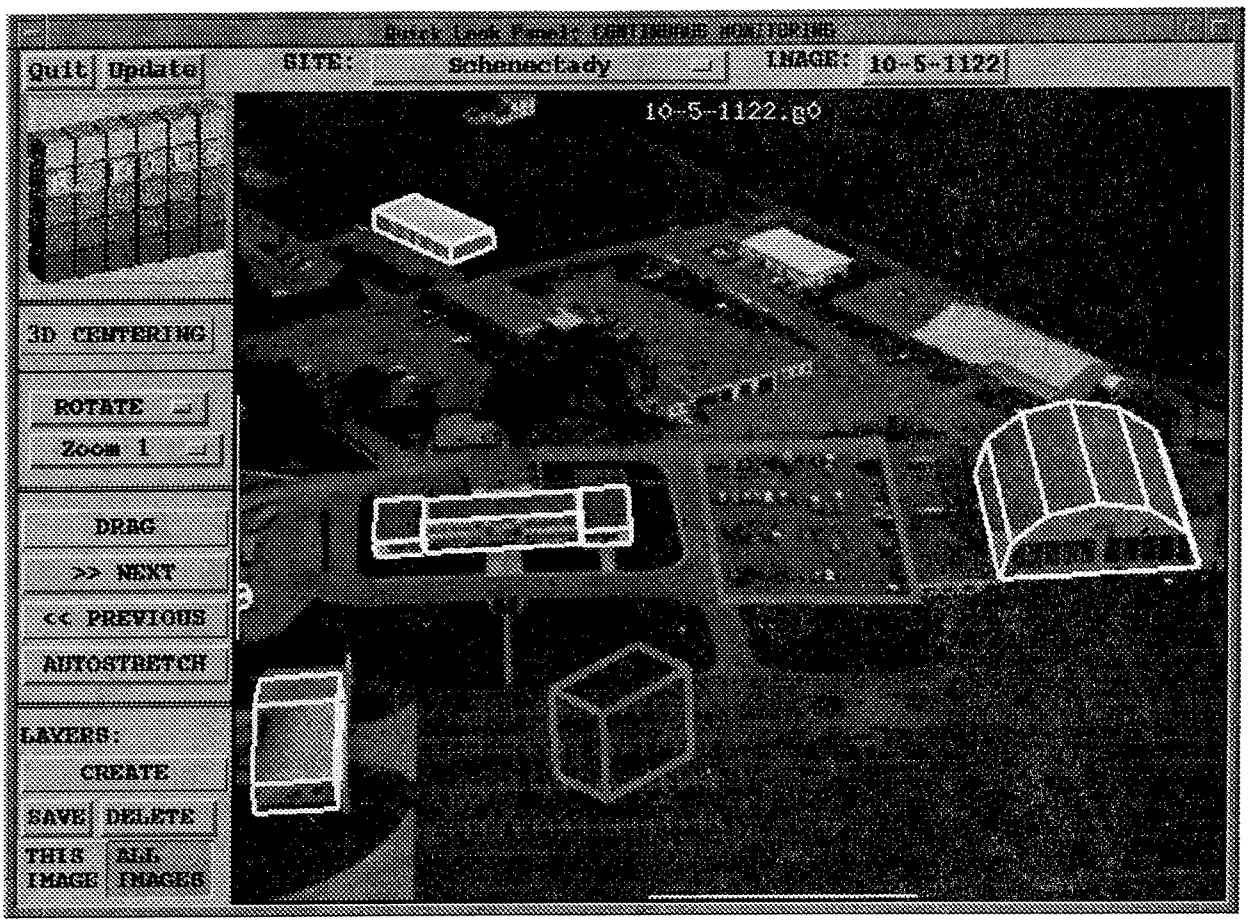


Figure 3: A typical quick look frame, prepared for the image analyst.

lar image. Both the shadow checking and the visibility checking algorithms were written by General Electric CRD.

Finally, another frequent problem affecting algorithm execution is cloud conditions. In aerial imagery, cloud conditions are a chronic problem. In a real-time scenario, an IA will not have time to review each image before execution is done. Instead, a two part cloud detection system was designed as part of the profiles system to prevent algorithm executions on imagery where the target region is cloud covered.

2 Visual Representation Of Algorithm Results

Once profile execution is complete, the IA must have an effective way of viewing the results. In the RADIUS testbed, every algorithm generates two important results. The first result an algorithm computes is a numeric change level, representing the amount of change found in a geometric object or region. This value is relative to the geometric model itself, or a previous image. The second result is the algorithm's confidence level. This number, scaled between 0.0 and 1.0, represents an algorithm's confidence in the results it has produced. Using these two factors, the RTS is able to make a boolean decision about whether change has occurred in a region.

2.1 Viewing Results

Once execution has completed and results are generated, visual representation of these results can be difficult. Given that each algorithm is responsible for producing its own change and confidence levels, consistency is a serious problem.

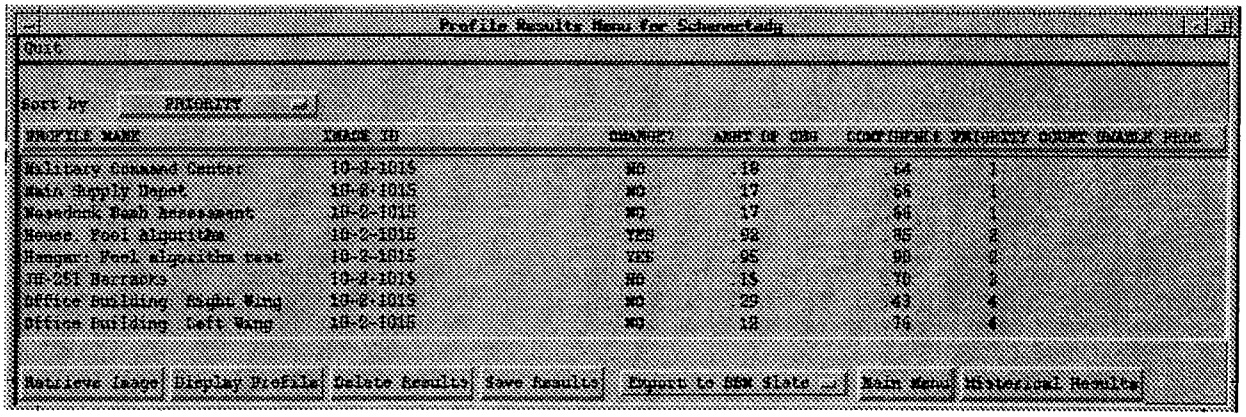


Figure 4: Sample profile results menu.

Rather than limiting our choices, the RTS has been implemented so that results are presented to the user in a variety of ways. The first way is known as quick look[4].

2.1.1 The Quick Look Display

In the intelligence community, a typical IA may have to examine a large amount of different imagery in a very short time frame. In the quick look paradigm, an IA is able to quickly scan an image previously processed by IU, and determine whether or not change has occured in his region of interest.

This is done by representing each structure or geometric region through a simple color scheme; red represents change, green represents no change, and light blue represents an ambiguous result. Through quick look, an IA can identify trouble regions in an image very quickly, saving him time and frustration. Figure 3 shows a sample quick look image. This image was the result of a series of *building validation* algorithms. Each wire frame outline in a quick look image represents a profile result. In this example, one building was flagged as changed, while no change was found in the four other structures.

2.1.2 Analyzing Algorithm Results

While the image based quick look system is available for fast, boolean results, many times it becomes necessary to analyze results in more depth. This can be done by using the profile results menu, as shown in Figure 4.

The profile results menu displays detailed information about each individual result. The list of results is pre-sorted by priority level, which was set during the profile creation phase. The IA may also sort the results on a variety of other values.

Each result is displayed in the results menu with corresponding values for the image identifier, amount of change, confidence level, priority, count (if applicable), and change boolean. There is also an "Unable to process" field, which would only be populated if the result was unable to run (in cases such as cloud cover or shadowing conditions).

2.1.3 Display of Historical Results

In order for an IA to accurately evaluate a region of interest, it may become necessary to view the results of previous executions on the same region. Through the RTS Database[5], historical results can be accessed and assessed. These historical results can be viewed in combination with current data to give the IA historical trends. Through the RTS, the IA may graph these trends using either a bar graph or a line graph. An example of the historical results menu is shown in Figure 5. The line graph in this figure represents the amount of change the IU found on a storage warehouse feature,

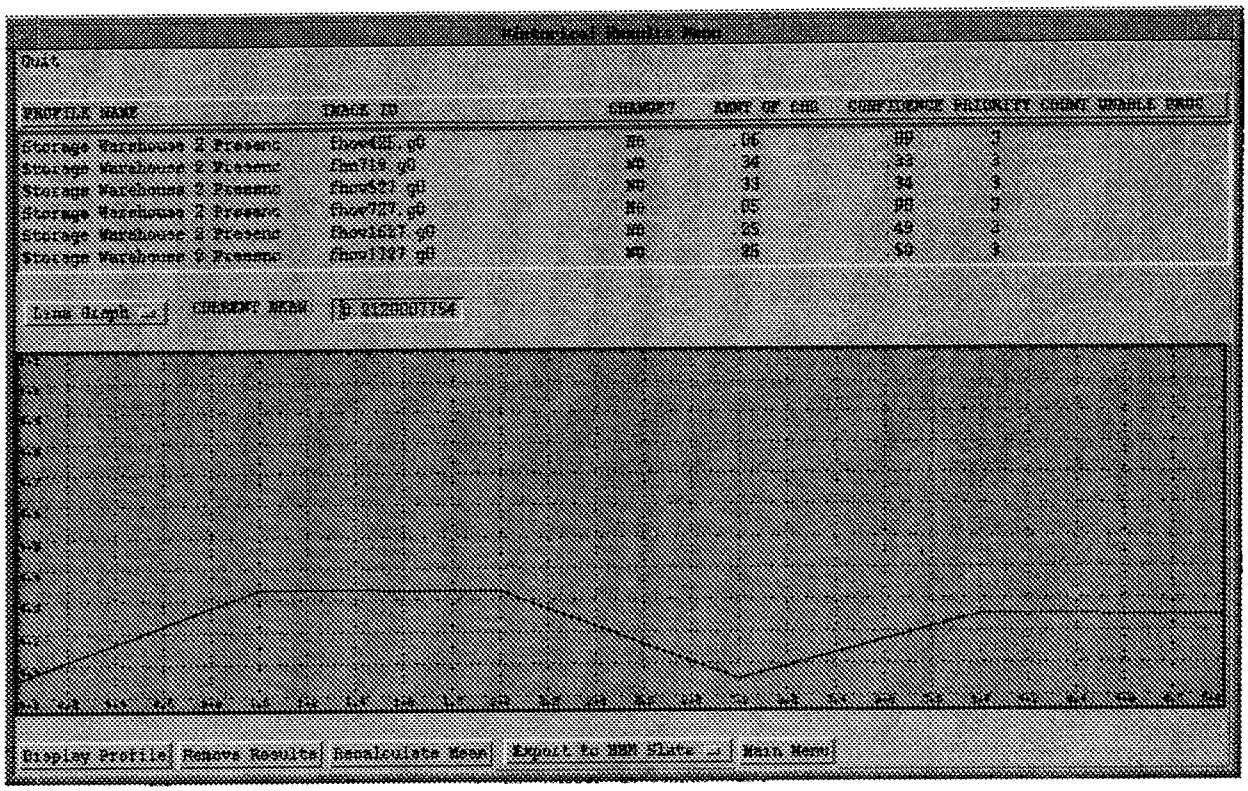


Figure 5: Sample profile results menu, showing histogram information.

graphed chronologically.

2.2 Interpretation of Change

The biggest dilemma in displaying profile results to the user is the question of how to visually represent change. When considering representation of change, one must also take the algorithm's confidence level into consideration. An algorithm that returns a high level of change with a low level of confidence may not be valuable to the IA.

This problem is compounded by the fact that each individual algorithm is responsible for generating its own confidence level. Some algorithms grade themselves harshly, while others are very liberal.

In the RTS, it was determined that an IA should have the ability to pre-set a minimum confidence level for each profile during the profile creation phase. Approaching the problem in this manner solves two problems.

First, results that fall below the confidence expectations of the IA are each flagged as an *ambiguous result* . This gives the IA the flexibility to investigate the result as he deems appropriate.

Secondly, the ability to set a minimum confidence level on a per algorithm basis allows the IA to evaluate the confidence level of each profile on a per algorithm basis. Algorithms that are known for being liberal in their confidence level grading can be pre-set to a higher confidence level for more rigorous screening. Algorithms that fail to meet expectations with regard to confidence level are flagged in light blue in the quick look display.

Finally, an IA may also set a sensitivity level for a feature profile. This value represents the level of change deemed to be significant, allowing pre-algorithm tailoring and giving the IA control of the significance of flagged results.

3 Conclusions

A significant challenge on RADIUS was to develop a consistent, friendly interface to IU systems that were inherently complex. To achieve this goal, feedback was received from a variety of sources. Lockheed Martin received input from our sponsor, the programming staff, and most importantly, image analysts, themselves.

In creating a user friendly interface, it was decided that a common interface was the best way to achieve familiarity. In pushing toward this goal, we have designed and implemented a common IU framework that rarely diverges to algorithm specific elements.

The RTS profiles system also uses a top-down design, and avoids putting too many buttons on a single menu. Multiple buttons and widgets on menus confuse and frustrate the user, reducing his efficiency. Alternatively, a top-down menu approach allows the user to proceed down a menu from beginning to end. Menu buttons are kept at a minimum, to reduce menu cluttering.

Finally, the quick look user interface system allows image analysts to see IU algorithm results in a quick and efficient manner. This is perhaps the RTS profiles system's greatest achievement in increasing an image analyst's efficiency and ability to do his everyday job.

4 Acknowledgments

We would like to thank the NIMA, specifically Ken Unger, for assisting with the user interface design of the RADIUS profiles system. In addition, we are grateful to our sponsors at DARPA and other government organizations for their support of RADIUS. Their support, personal interest, and encouragement helped make the RADIUS program a success. We also wish to thank all LMC contributors to this work, especially Bethany Kniffin, Bill Bremner, and Doug Hackett.

References

[1] J. Sargent, M. Kelly, and J. Baily, "RADIUS Concept Definition Experiments," *Proceedings of the ARPA Image Understanding Workshop* , Nov. 1994.

[2] "Human-Computer Interface (HCI) Specification for the RADIUS Testbed System." Prepared by the National Exploitation Laboratory, August 1995.

[3] B. Bremner, A . Hoogs and J. Mundy. "Integration of Image Understanding Exploitation Algorithms into the RADIUS Testbed," *Proceedings of the ARPA Image Understanding Workshop* , Feb. 1996.

[4] J. Baily, M. Kelly, and J. Sargent. "Quick Look: A New Way to Prioritize Imagery for Exploitation," *Proceedings of the ARPA Image Understanding Workshop* , Nov. 1994.

[5] B. Kniffin and A. Hoogs. "Database Support for Exploitation Image Understanding," *Proceedings of the ARPA Image Understanding Workshop* , Feb. 1996.

[6] Martin Marietta and SRI International, "RCDE User's Guide", Martin Marietta Management and Data Systems, Philadelphia, PA 1993.

A Geometric Framework for Image Alignment

Venu Govindu Chandra Shekhar Rama Chellappa
Center for Automation Research, University of Maryland
College Park, MD 20742-3275

Abstract

In this paper we introduce a framework for correspondence-less planar image alignment using global geometric descriptors of image primitives. The alignment is achieved by appropriately parametrizing the required transformation between images and estimating these parameters. The parameters are estimated by comparing the aggregate descriptors of the geometric properties of primitives in the two images. This comparison is carried out in an estimation-theoretic framework. We define the concepts of parameter observability and separability which are used to guide the choice of geometric descriptors. The method proposed in this paper has a wide range of applications including multi-sensor image registration, mosaicking and pose estimation. Examples of experiments on real data are provided.

1 Introduction

The alignment of two images amounts to placing the images in a common frame of reference. The requirement of alignment arises in a wide range of applications including multi-sensor data fusion, change detection, pose recovery and object recognition. For many such image understanding applications analysis is possible only if the image data are co-aligned, or "registered", with respect to a common coordinate system.

Most methods of alignment (also referred to as registration or positioning) are characterized by their choice of a feature space and the similarity metric used to determine the geometric transformation required for alignment. This is typically achieved by modeling the geometric transformation between the two images and estimating the transformation using information common to both images. The estimated transformation is then used to bring one image into alignment with the other. Image alignment has been widely investigated and a vast amount of literature is available. A good survey can be found in [Brown 92].

Typically, the transformation is computed using either a feature-matching technique [Li *et al.* 95] or a search strategy [Viola and Wells 95; Fua and Leclerc 94]. Feature-based methods traditionally rely on establishing a feature correspondence between the two images. Such correspondence-based methods first employ feature-matching techniques to determine corresponding feature pairs from the two images and then compute the geometric transformation relating them, typically using a least-squares approach. Their primary advantage is that the transformation parameters can be computed in a single step and are accurate if the feature matching is reliable. Their drawback is that they require a heuristic method of feature matching that is specific to the domain of the problem. The problem is further compounded in the multi-sensor scenario where the common or "mutual" information may manifest itself in a different manner for each image since different sensors record different physical phenomena in the scene. For instance, an infra-red sensor responds to the temperature distribution of the scene, whereas a radar responds to material properties such as dielectric constant, electrical conductivity, etc. Feature matching is also computationally expensive due to the well-known *correspondence problem* (given N features in each image, the number of possible one-to-one feature mappings is $N!$, out of which only one is correct). Heuristics can be employed to reduce the number of potential mappings, but the problem still remains intractable, unless the two images are already approximately aligned or the number of features is small.

Search strategies are difficult to characterize, suffer from the problems of local minima and require good

initial guesses. While there has been some work on alignment without correspondence, it either deals with alignment of a single object (e.g. [Kumar and Goldgof 96]), needs good initial guesses (e.g. [Fua and Leclerc 94]) or recovers only translation [Basu and Aloimonos 90].

However, since the underlying scene giving rise to the shared information is the same, certain geometric properties are preserved across multi-sensor data. Although the corresponding pixels may have different values, *similarity* and *dissimilarity* of pixel groups or regions is usually preserved. Regions are either homogeneous or can be easily distinguished in either image. Also man-made objects in a scene such as buildings and roads in aerial imagery and implants, prostheses, metallic probes, etc. in medical imagery give rise to features that are likely to be preserved in multi-sensor images.

The methodology proposed in this paper alleviates these limitations by making fewer implicit assumptions. Also, the use of global distributions of geometric properties makes this method more robust with regard to problems of occlusion, clutter and errors in low-level processing. Since the geometric properties of image primitives like points, lines, curves, etc. remain relatively stable, they are often sufficient to determine the transformation between the images. In the remainder of this paper we shall assume that an image consists of a collection of such primitives. In Section 3, we describe how alignment can be achieved by relating the global distributions of specific geometric properties of the two images. This is done without explicit feature matching or searching over a multi-dimensional parameter space. Instead, we relate the geometric properties of one image to those of the other via the transformation parameters. This allows us to build global distributions of the geometric properties which can then be used to align the two images.

The remainder of this paper is organized as follows. Section 2 defines the problem of image alignment and the solution framework is described in Section 3. In Section 4 we show the solutions for specific transformation models. In Section 5 we present some results and finally we discuss issues related to our method in Section 6.

2 Problem Definition

The scene being imaged is considered to be embedded in a plane (denoted by Ω) and the image is assumed to be generated by viewing this scene using sensor f_k. Let us denote

$$I_i = T_i(\Omega, P_i, f_k),\ T_i : \mathcal{R}^2 \longrightarrow \mathcal{R}^2$$

where I_i denotes the ith image of a scene Ω using the kth sensor and P_i is the relative sensor orientation. The images can be modeled as follows:

Consider a pair of images I and $\tilde{I}$ obtained by imaging the scene Ω. If Ω_1 and Ω_2 are portions of the scenes being imaged, we have

$$I = T_1(\Omega_1),\ \Omega_1 \subset \Omega$$
$$\tilde{I} = T_2(\Omega_2),\ \Omega_2 \subset \Omega$$

where

$$\Omega_1 \cap \Omega_2 \neq \emptyset$$

Without loss of generality, we can designate I as the "base frame" (i.e. frame of reference). Thus the problem of image alignment can be stated as follows:

Given images I and $\tilde{I}$ of Ω, compute the composite transformation $T = T_1 \circ {T_2}^{-1}$ such that

$$T : T_2(\Omega_1 \cap \Omega_2) \rightarrow T_1(\Omega_1 \cap \Omega_2).$$

In practice, the form of T is chosen based on experience and knowledge of sensor geometries and the nature of the scene.

3 Geometric Framework

As mentioned in Section 1, the primitives in different images can be of various types, such as points, lines, edges, curves, regions, etc.* We assume the images to be composed of collections of geometric primitives $\mathbf{p}$ of a given type k. Therefore

$$I = \bigcup_i \left\{ \mathbf{p}_i^{(k)} \right\}$$

Henceforth we shall drop the superscript k for convenience. Every primitive has geometric properties associated with it. Typical properties are position, slope, curvature, length, area, etc. We define the following in order to study the relationship between primitive properties and transformation parameters.

Definition 1 *$(D, \tilde{D})$ is an operator pair such that*

$$D(\mathbf{p}) = \tilde{D}(\tilde{\mathbf{p}}, g(T)), \forall\ (\mathbf{p}, \tilde{\mathbf{p}}),$$

where $\mathbf{p}, \tilde{\mathbf{p}}$ are image primitives in images $I, \tilde{I}$ respectively and $g(T)$ is a function of the transformation T.

If the transformation T is characterized by n parameters $a_1, a_2, \cdots, a_n$, then the function $g(.)$ is also a function of these n parameters. If the function $g(T)$

*The types of primitives present are typically determined by the nature of the scene being imaged. For example, lines would be predominant in an urban or industrial scene whereas curves would be predominant in medical images or natural scenes.

is of a known form and can be recovered from the operator pair $(D, \tilde{D})$, then $g(T)$ is said to be *observable*. Hence to be able to compute the transformation $T = T_{a_1,a_2,\cdots,a_n}$ we need n operator pairs that make observable either the parameters $\{a_1, a_2, \cdots, a_n\}$ or $\{g_1, g_2, \cdots, g_n\}$ which are n independent functions of $\{a_1, a_2, \cdots, a_n\}$ such that $T = T_{g_1,g_2,\cdots,g_n}$.

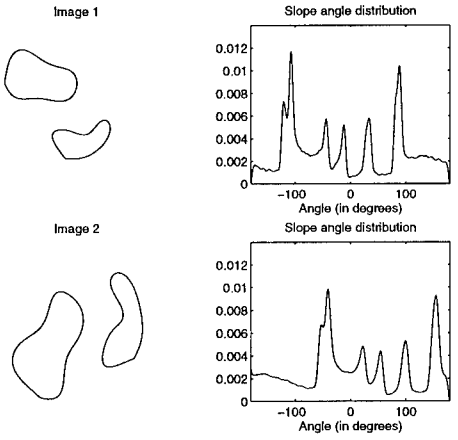


Figure 1: Parameter observation through distributions. Image 2 is a Euclidean transformed version of Image 1.

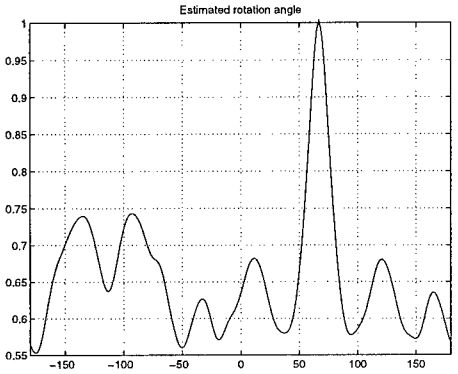


Figure 2: The peak is the estimated angle of rotation in degrees.

As stated above, we can relate the geometric properties of image primitives in the two images and thereby recover the transformation between them. Consider a primitive $\mathbf{p}$ in image I and its corresponding primitive $\tilde{\mathbf{p}}$ in image $\tilde{I}$. The geometric properties of $\mathbf{p}$ and $\tilde{\mathbf{p}}$ are related to each other through the transformation T. For example, in Fig. 1 we have an image and its Euclidean transformed version. The slope angles at any corresponding point pair $(\mathbf{p}, \tilde{\mathbf{p}})$, as defined above, would be related through the rotation angle θ. As can be observed in Fig. 1, the two distributions can be used to recover the rotation angle θ (see Fig. 2) by simply computing the shift between the slope angle distributions of the two images. It may be noted that in such a scenario we do not require *any* knowledge of matches of features or primitives. Instead the distributions are computed in an independent fashion using each image. Such a notion of parameter estimation by comparing global distributions of geometric properties can be extended to other parameters of the transformation. To formalize this notion we now need to define the concept of a descriptor set and its corresponding probability function.

As before, $(\mathbf{p}, \tilde{\mathbf{p}})$ denote corresponding primitives in the two images I and $\tilde{I}$ respectively.

Definition 2 *$\mathcal{M}(I) = \{D(\mathbf{p}) | \forall \mathbf{p} \epsilon I\}$ and $\tilde{\mathcal{M}}(\tilde{I}) = \{\tilde{D}(\tilde{\mathbf{p}}) | \forall \tilde{\mathbf{p}} \epsilon \tilde{I}\}$ are defined to be the descriptor sets of images I and $\tilde{I}$ respectively. $\mathcal{P}(\mathcal{M})$ and $\mathcal{P}(\tilde{\mathcal{M}})$ denote the probability measures of the descriptor sets $\mathcal{M}$ and $\tilde{\mathcal{M}}$ respectively.*

The Maximum Likelihood Estimator (MLE) for the parameter $g(T)$ can be stated as follows :

Given an image pair I and $\tilde{I}$, and an operator pair $(D(\mathbf{p}), \tilde{D}(\tilde{\mathbf{p}} | g(T)))$, the Maximum Likelihood Estimator (MLE) for $g(T)$ is

$$\arg \min_{g(T)} \| \mathcal{P}(\mathcal{M}) - \hat{\mathcal{P}}(\tilde{\mathcal{M}} | g(T)) \|_n \qquad (1)$$

where $\mathcal{P}$ and $\hat{\mathcal{P}}$ are the observed probability measures of $\mathcal{M}$ and $\tilde{\mathcal{M}}$ respectively and $\| \cdot \|_n$ is the L_n norm for some $n > 0$. The probability distribution function (pdf) of the observation noise is assumed to be monotonically decreasing with a mode at zero.

In cases when the parametrization is of a form such that the descriptors are linearly related, the observed parameter is said to be *separable*. For example, under the Euclidean subgroup of transformations, if the parameter to be estimated is the rotation angle (denoted by $\Delta\theta$), then as described above the operator pair is defined to be the slope angles of points on lines or curves, i.e. $(D, \tilde{D}) = (\theta, \tilde{\theta})$. Here the relationship can be simply expressed as

$$\theta = \tilde{\theta} + \Delta\theta$$

This separability allows the use of fast methods like cross-correlation to compute the observed parameter $\Delta\theta$.

Often it is easier to impose an ordering on the computation of the parameters which simplifies subsequent computational stages. For example, under similarity, it is easier to compensate for scaling and rotation before computing translation. It must be noted that the operators have to be chosen in a manner that is independent of the parameters of T not accounted for by $g(T)$. For example, to compute scale under a similarity transformation, we can use

the radius of curvature as an operator, since it is independent of the rotation and translation parameters (Subsection 4.2). However, such a separation of the parameters may not always be possible (e.g., under the affine group of transformations). In that case, the transformation has to be reparametrized in such a way that we can independently estimate the new parameters.

An important point to note is that the proposed method is *not* a stochastic minimization technique. As will be shown in the following section, the transformation can be reparametrized such that $g(T)$ has a finite range, and since all descriptor measures are finite, the distribution functions have finite support. This in turn implies a fixed amount of computation at each stage of estimation.

The geometric properties used to compute the parameters depend on the types of image primitives. For continuous curves we use differential properties, whereas for discrete points and lines the corresponding geometric descriptors are discrete in nature. In the sections that follow we shall give examples of the use of both discrete and differential-geometric properties to recover the parameters under observation.

4 Transformation Models

Throughout this section we shall denote the points in the first image by $\mathbf{p} = (x, y)^T$ and those in the second image by $\tilde{\mathbf{p}} = (\tilde{x}, \tilde{y})^T$. In the following subsections, we shall develop solutions for some transformation models. It may be noted that the points are associated with specific image primitives like image points, lines, curves, etc. The transformation model is

$$\tilde{\mathbf{p}} = T\mathbf{p} + \mathbf{t} \tag{2}$$

where $\mathbf{t} = (t_x, t_y)^T$ is the 2-D translation vector and the matrix T is a 2×2 invertible matrix. Henceforth, matrix T will be referred to as the transformation matrix. Using curves as primitives will be referred to as the differential case and using points and lines will be referred to as the discrete case.

4.1 Euclidean

A simple 2-D transformational subgroup is the Euclidean model consisting of a rotation (θ) and two translational parameters (t_x, t_y). Here we have

$$\tilde{\mathbf{p}} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \mathbf{p} + \mathbf{t} \tag{3}$$

In the differential case, to compute rotation we choose the operators $(D(\mathbf{p}), \tilde{D}(\tilde{\mathbf{p}}))$ to be the slope angles of points of the curves in the two images. For the example shown in Fig. 1, the rotation value can be computed using the MLE defined above. In the discrete case the rotation parameter is observable through the slope angles of lines. It may be noted that in this case, the function $g(T)$ takes on the simple form $g(T) = \theta$. Having compensated for the rotation between the images, we can compute the x-direction translation t_x between the two images using $D(\mathbf{p}) = x(\mathbf{p})$ and $\tilde{D}(\tilde{\mathbf{p}}) = x(\tilde{\mathbf{p}})$, where $x(\mathbf{p})$ is the x-coordinate of point $\mathbf{p}$. The y-component of translation can be computed in a similar fashion.

4.2 Similarity

To compute the similarity transformation, we need to compute an additional scale parameter s. Recalling that the radius of curvature (R) of a point on a curve is directly proportional to the scaling parameter, we have $R(\mathbf{p}) = sR(\tilde{\mathbf{p}})$. From this we can deduce that $D(\mathbf{p}) = \ln(R(\mathbf{p}))$ and $\tilde{D}(\tilde{\mathbf{p}}) = \ln(R(\tilde{\mathbf{p}}))$ in the differential case. In the discrete case we can use the distance (d) between a pair of points instead of the radius of curvature, in which case $\tilde{d} = sd$. Hence we have the simple additive relationship

$$D(\mathbf{p}) = \tilde{D}(\tilde{\mathbf{p}}) + \ln(s)$$

It may also be noted that we can compute the scaling and rotation parameters of the transformation independently.

4.3 Quasi-affine

We now look at the case of a quasi-affine transformation which we define as

$$\tilde{\mathbf{p}} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \mathbf{p} + \mathbf{t} \tag{4}$$

In this case, the curvature and slope angles are no longer independent. However we can reparametrize (4) as

$$\tilde{\mathbf{p}} = \sqrt{|s_x s_y|} \begin{pmatrix} \rho & 0 \\ 0 & \frac{1}{\rho} \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \mathbf{p} + \mathbf{t}\,, \tag{5}$$

where $\rho^2 = \frac{s_x}{s_y}$.

To compute the determinant of the transformation ($|T| = |s_x s_y|$) in the differential case, let us assume that the curves are parametrized according to their arc-length representations [Bruckstein *et al.* 93]. Let us denote by s and $\tilde{s}$ the arc-length indices of the curves in images I and $\tilde{I}$ respectively. We define the 2×2 matrices P and $\tilde{P}$ as $P = [\dot{\mathbf{p}}(s), \ddot{\mathbf{p}}(s)]$ and $\tilde{P} = [\dot{\tilde{\mathbf{p}}}(\tilde{s}), \ddot{\tilde{\mathbf{p}}}(\tilde{s})]$ for point pairs $(\mathbf{p}, \tilde{\mathbf{p}})$, where the dots denote derivatives with respect to the arc-length parametrization of the curve.

From the definition of the transformation, we get

$$\tilde{P} = TP \Rightarrow \ln|\tilde{P}| = \ln|s_x s_y| + \ln|P|$$

Therefore we use $D(\mathbf{p}) = \ln|[\dot{\mathbf{p}}, \ddot{\mathbf{p}}]|$ and $\tilde{D}(\tilde{\mathbf{p}}) = \ln|[\dot{\tilde{\mathbf{p}}}, \ddot{\tilde{\mathbf{p}}}]|$ to compute the determinant of the transformation T. It is important to observe that in the above analysis, we do not actually need to compute an "arc-length" reparametrization of the curves that is invariant to the transformation. It suffices to compute the descriptors for a sufficiently dense set of points on the curves. This can be achieved by using local fitting of curves ([Bruckstein *et al.* 93], for example). By scaling the curves we get the new relationship

$$\dot{\tilde{\mathbf{p}}} = \begin{pmatrix} \rho & 0 \\ 0 & \frac{1}{\rho} \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \dot{\mathbf{p}} \qquad (6)$$

The rotation angle θ can be computed in a manner invariant to the ratio of the scales ρ^2. To do this we use the relationship

$$\dot{\tilde{x}}\ddot{\tilde{y}} = \frac{\sin(2\theta)}{2}(\dot{y}^2 - \dot{x}^2) + \cos(2\theta)\dot{x}\dot{y} \qquad (7)$$

The two sides of (7) can be equated to $D(\mathbf{p})$ and $\tilde{D}(\tilde{\mathbf{p}}, \theta)$ and θ can be solved for.

The parameter ρ, now satisfies the relationship

$$|\frac{\dot{\tilde{x}}}{\dot{\tilde{y}}}| = |\rho^2|\,|\frac{\cos(\theta)\dot{x}+\sin(\theta)\dot{y}}{-\sin(\theta)\dot{x}+\cos(\theta)\dot{y}}|, \qquad (8)$$

which implies that

$$ln|\frac{\dot{\tilde{x}}}{\dot{\tilde{y}}}| = 2ln|\rho| + ln|\frac{\cos(\theta)\dot{x}+\sin(\theta)\dot{y}}{-\sin(\theta)\dot{x}+\cos(\theta)\dot{y}}| \qquad (9)$$

Thus we can also solve for the parameter ρ and hence we can recover the transformation matrix T.

In the discrete case we similarly note that using a pairs of lines as the primitives, the rotation angle (θ) is observable but not separable. Consider a unit vector $\mathbf{v}$ at an angle ϕ, given by

$$\mathbf{v} = \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix}$$

Under the transformation it becomes

$$\tilde{\mathbf{v}} = \begin{pmatrix} \rho & 0 \\ 0 & \frac{1}{\rho} \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \mathbf{v}$$

$$\Rightarrow \tilde{\mathbf{v}} = \begin{pmatrix} (1/\rho)\cos(\phi+\theta) \\ \rho\sin(\phi+\theta) \end{pmatrix}$$

Let $\tilde{m} = \tan\tilde{\phi}$ be the slope of $\tilde{\mathbf{v}}$. By dividing the y-component of $\tilde{\mathbf{v}}$ by its x-component, we get

$$\tan\tilde{\phi} = \rho^2 \tan(\phi + \theta) \qquad (10)$$

In order to observe θ, we need to eliminate ρ from the above equation. This can be achieved by taking the ratio of two line slopes. Let lines with slope angles ϕ and ψ be transformed into lines with slope angles $\tilde{\phi}$ and $\tilde{\psi}$.Then

$$\frac{\tan\tilde{\phi}}{\tan\tilde{\psi}} = \frac{\tan(\phi+\theta)}{\tan(\psi+\theta)} \qquad (11)$$

Thus given line pairs from each image, the only unknown in (11) is the rotation angle θ. After some simple manipulations, we obtain the following expression for θ:

$$\theta = (1/2)\sin^{-1}(k\sin(\phi-\psi)) - \phi/2 - \psi/2$$

where

$$k = \frac{\tan\tilde{\phi} + \tan\tilde{\psi}}{\tan\tilde{\phi} - \tan\tilde{\psi}}.$$

After observing θ, image I can be rotated accordingly, and (10) simplifies to

$$\tan\tilde{\phi} = \rho^2 \tan\phi \qquad (12)$$

Using (12), ρ can be observed in a separable fashion.

Finally it may be noted that the scaling and translation between the two images can be determined for the differential and discrete cases as in the case of the similarity transformation.

4.4 Affine

In the more general case we have an affine transformation between two images which can be described by the equation

$$\tilde{\mathbf{p}} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mathbf{p} + \mathbf{t} \qquad (13)$$

where the transformation matrix T is a non-singular 2×2 matrix. The computation of the four independent parameters of the transformation matrix T can be accomplished in the differential case in the following manner.

The value of $|T|$ can be computed in a manner identical to that in the previous section. Thereafter, we need three more independent equations to solve for the transformation.

We can derive the simple relationship

$$\frac{\dddot{\tilde{x}}}{\dot{\tilde{x}}} = \frac{a\dddot{x} + b\dddot{y}}{a\dot{x} + b\dot{y}}$$

By a simple reparametrization of the parameters $(a, b) = \sqrt{a^2+b^2}(\sin(\phi), \cos(\phi))$ in the above equation, we get

$$\frac{\dddot{\tilde{x}}}{\dot{\tilde{x}}} = \frac{\sin(\phi)\dddot{x} + \cos(\phi)\dddot{y}}{\sin(\phi)\dot{x} + \cos(\phi)\dot{y}} \qquad (14)$$

Figure 3: On the left are some frames from a sequence of images. The mosaic of these images is shown on the right.

from which we can solve for the ratio a/b using the left-hand side of (14) for $D(\mathbf{p})$ and the right-hand side for $\tilde{D}(\tilde{\mathbf{p}}, g(T))$.

By similar analysis, we can recover the ratio c/d using the relationship for $\ddot{\tilde{y}}/\dot{\tilde{y}}$. Finally, since we now know the ratios a/b and c/d we can observe that

$$\frac{\dot{\tilde{x}}}{\dot{\tilde{y}}} = \left(\frac{b}{d}\right) \frac{\frac{a}{b}\dot{x} + \dot{y}}{\frac{c}{d}\dot{x} + \dot{y}}$$

from which the ratio b/d can be computed in a separable fashion. It may be noted that the reparametrization of (a, b) and (c, d) results in a finite range of possible estimate values of the new parameters (i.e., the range of ϕ is $[-\pi, \pi]$).

For the discrete case we adopt the following methodology. Using the QR transformation from linear algebra, the transformation can be written in terms of six parameters (rotation θ, translation t_x, t_y, scale ratio $\rho = \sqrt{\frac{s_x}{s_y}}$, scale $\Delta = \sqrt{|s_x s_y|}$ and skew α):

$$\tilde{\mathbf{p}} = \Delta \begin{pmatrix} 1/\rho & 0 \\ \alpha & \rho \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \mathbf{p} + \mathbf{t} \tag{15}$$

Proceeding as in the previous case, it can be shown that the rotation angle is observable using triplets of lines as primitives. If a primitive pair consists of lines at slope angles (ϕ, ψ, λ) and $(\tilde{\phi}, \tilde{\psi}, \tilde{\lambda})$, we can show that the rotation angle that is consistent with the feature pair is given by

$$\theta = \tan^{-1}\left(\frac{-\cos\lambda + k\cos\psi}{\sin\lambda - k\sin\psi}\right)$$

where

$$k = \frac{(\tan\tilde{\phi} - \tan\tilde{\psi})}{(\tan\tilde{\phi} - \tan\tilde{\lambda})} \frac{\sin(\phi - \lambda)}{\sin(\phi - \psi)}$$

Once rotation is compensated for, the scale ratio is separably observable using line pairs as features, according to

$$\tan\tilde{\phi} - \tan\tilde{\psi} = \rho^2(\tan\phi - \tan\psi)$$

After compensating for the scale ratio, the skew is separably observable from line slopes, according to

$$\tan\tilde{\phi} = \tan\phi + \alpha$$

The scale and translation are determined as in the previous cases.

5 Results

In this section we present the results we have obtained on several sets of images. The use of both discrete and differential-geometric properties is illustrated in this section. The choice of image primitive depends on the type of images being aligned. In practice, urban areas have mostly straight lines and hardly any curves. In such cases it is difficult to compute the differential properties whereas discrete properties can be easily extracted. The converse applies for natural scenes where straight edges are seldom available. The left half of Fig. 3 shows some of the frames from a sequence of images. The right half of the same figure shows a mosaic generated using the sequence. As can be observed there is little or no overlap between images that are well separated in the sequence. However, since there is overlap between adjacent frames, we can register all the images to a common frame of reference, thereby generating a panoramic view of the scene.

Fig. 4 shows two aerial images of a scene obtained by using sensors that operate in different ranges of the electro-magnetic spectrum. The image on the left was obtained by the MODIS Airborne Simulator

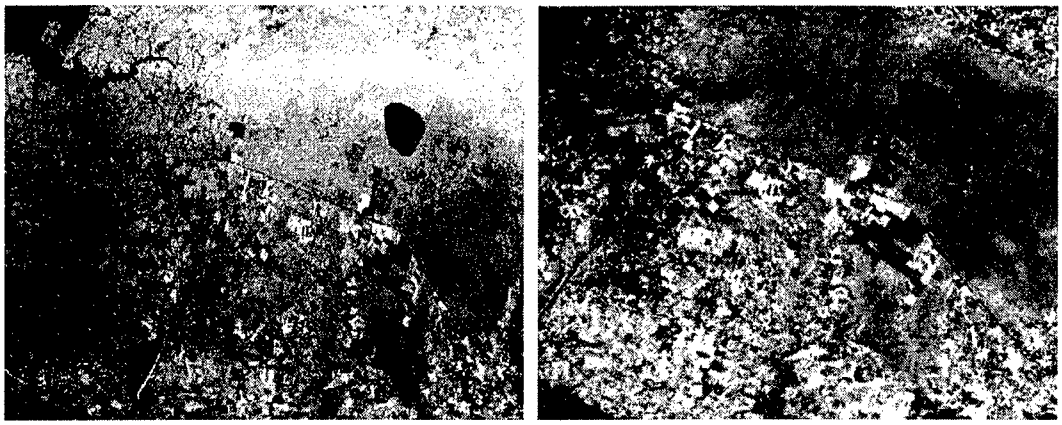


Figure 4: Aerial images obtained using different sensors. The image on the left is an MAS image and the one on the right is a TM image.

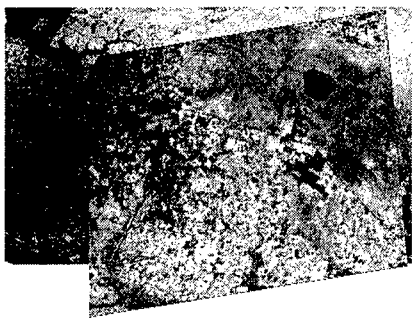


Figure 5: Overlayed aerial images: The TM image of Fig. 4 is overlayed on the MAS image of the same figure.

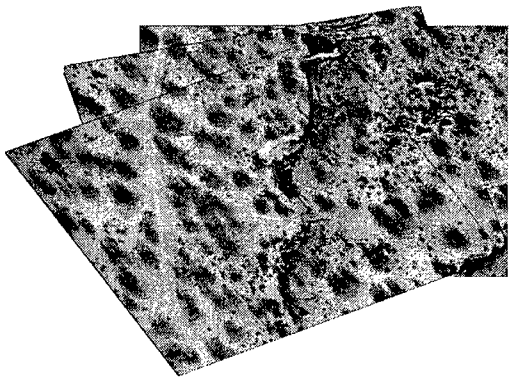


Figure 6: Alignment of a set of aerial images using a similarity model.

(MAS) and the corresponding image on the right is one of the bands of the Thematic Mapper (TM). Fig. 5 shows the TM image overlayed on the MAS image. A quasi-affine model was used for computing the transformation between the two images.

Fig. 6 shows the registration of images of the Mojave desert obtained from a balloon flying over the region. Adjacent pairs of images were registered and the images were aligned in a common reference frame. In this case the similarity model was sufficiently accurate. The registered image set shows the correct alignment of features like roads, rock outcrops, etc. in spite of the presence of non-overlapping structure in the image pairs.

As a final example of the differential case, we show the registration of two MRI images (Fig. 7). The image on the left in Fig. 7 is a proton density MRI image and the one in the middle is the corresponding T2-weighted image with an arbitrary alignment. In the image on the right we show the alignment using an affine model of one pair of the contours from the two different modalities.

We illustrate the performance of the discrete method on a multi-sensor data set, containing a SAR-visual image pair (Fig. 8) with viewpoint and photometric differences. The final result after applying all the stages of transformation is shown by overlaying the contours of the visual scene on the SAR image.

All the above examples demonstrate the effectiveness of image alignment using our method.

6 Discussion

A significant advantage of our methodology is its applicability to a wide variety of scenarios. This is possible since we use the same distributional framework of geometric properties to estimate all the parameters. Moreover the use of a distributional approach provides robustness with respect to errors in low-level processing. This results in a graceful degradation in parameter estimation with increased occlusion, which is desirable. In fact, with enough dominant structure in the images, we have observed very

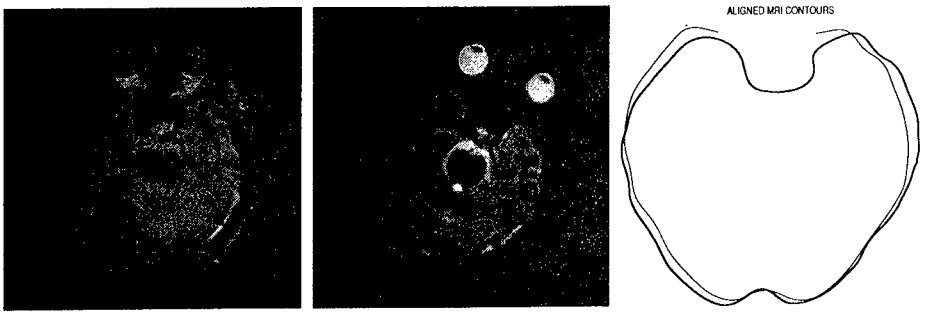


Figure 7: MRI image registration : The image on the left is a proton density image, the one in the middle is T2-weighted, and the image on the right shows the registration of a single contour of the two modalities.

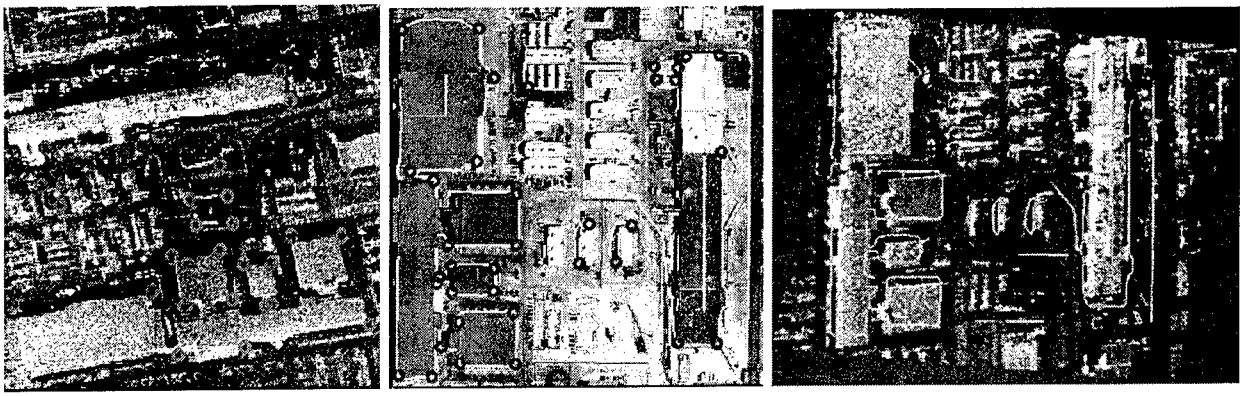


Figure 8: The leftmost and center images are SAR and visual images, respectively. The rightmost figure shows the registration achieved, by overlaying the visual contours on the SAR image.

little degradation in performance in the presence of occlusion.

Since we use one-dimensional comparisions as our means of parameter estimation, our method is significantly faster than voting schemes that are multi-dimensional, such as the Hough transform, etc. Also, unlike the Hough scheme, we do not "hypothesize" any matching pairs; instead, all computations are carried out independently on individual images. Such a separation of information is very useful since it allows us to use better comparisions as compared with the "blind" methods of multi-dimensional voting schemes. For example, once we have determined the appropriate rotation for a similarity model, the fact that we have compensated for the rotation can be exploited during the computation of the scale. Thus the comparision metric can be modified in such a way that we do not "hypothesize" pairings of primitives that have vastly different slope values. Such a progressive filtering of the possible combinations has proven to be very useful in scenarios with significant amounts of non-overlap. This has resulted in more robust estimation of the relevant transformation parameters.

Interestingly, a straightforward comparison of the functional similarity of distributions can determine the extent of image overlap. Also, while the true MLE is unique, in practice multiple hypotheses (peaks) arise due to the presence of clutter. Such a scenario often occurs when dealing with urban scenes which often contain rectangular structures (buildings, etc.) that give rise to spurious peaks that are 90 degrees away from the true rotation maxima. Under such circumstances, disambiguation of the hypotheses can be achieved by using a measure of quality that is independent of the parameter being estimated. Another method is to use a tree descent strategy that combines the information extracted at different stages to disambiguate between multiple hypotheses.

Finally, it may be noted that in certain degenerate situations the true MLE may not be *visible*.[†] Under such situations the "flatness" of the descriptor distributions indicates the non-existence of a solution since all hypotheses are equally likely.

As our examples illustrate, the theoretical framework we have developed performs well for real images. Moderate amounts of clutter and occlusion do not affect the geometric properties of the primitives

[†]For illustrative purposes, consider the rotation of a circle. The distributions of slope angles are uniform, indicating the non-existence of a unique MLE.

that are common to both images. Hence the descriptor distributions do effectively "capture" the information relevant to parameter estimation. The same applies for images with small amounts of overlap. Under such circumstances we have noted that an iterative refinement of the transformation estimate is useful. The estimates are used to "window" the images so as to increase the fraction of overlap and the parameters are estimated again. In many cases the solutions converge in a few iterations. This is generally true unless the distributions are "extremely" corrupted due to the presence of clutter and non-overlapping components.

A limitation of our method is the requirement of a planar scene, an assumption which would be violated in the presence of large perspective effects due to three-dimensional structure. We need to investigate the extension of our method to the non-planar case. However, if the perspective effects are tolerable, our approach can be used to get an initial alignment.

7 Conclusion

In this paper we have described a framework for alignment of images without the explicit use of correspondence. We have demonstrated the effectiveness of using a distributional description of geometric properties of images in achieving alignment. We argue that while feature correspondence is generally intractable for many situations, especially multi-sensor image pairs, our method can very effectively handle such scenarios due to the use of image properties that are relatively invariant to changes due to multiple sensors, change in illumination direction, etc. Also a sequential estimation of one-dimensional parameters results in faster and more robust estimation compared to multi-dimensional voting schemes. We are currently investigating methods of extending the domain of applicability of our method and of further refining the estimation process.

References

[Basu and Aloimonos 90] A. Basu and Y. Aloimonos, "A Robust, Correspondenceless, Translation-Determining Algorithm", *International Journal of Robotics Research*, Vol. 9, No. 5, pp. 35-59, 1990.

[Brown 92] L. G. Brown, "A Survey of Image Registration Techniques", *ACM Computing Surveys*, Vol. 24, No. 4, pp. 325-376, 1992.

[Bruckstein *et al.* 93] A. Bruckstein, R. Holt, A. Netravali, and T. Richardson, "Invariant Signatures for Planar Shape Recognition Under Partial Occlusion", *CVGIP: Image Understanding*, Vol. 58, No. 1, pp. 49-65, 1993.

[Fua and Leclerc 94] P. Fua and Y. Leclerc, "Image Registration Without Explicit Point Correspondences", *Proceedings of DARPA IUW*, pp. 981-992, 1994.

[Kumar and Goldgof 96] S. Kumar and D. Goldgof, "Recovery of Global Nonrigid Motion—A Model-Based Approach Without Point Correspondences", *Proceedings of CVPR*, pp. 594-599, 1996.

[Li *et al.* 95] H. Li, B. Manjunath and S. Mitra, "A Contour-Based Approach to Multisensor Image Registration", *IEEE Transactions on Image Processing*, Vol. 4, No. 3, pp. 320-334, 1995.

[Viola and Wells 95] P. Viola and W. Wells, "Alignment by Maximization of Mutual Information", *Proceedings of ICCV*, pp. 16-23, 1995.

Multiple View 2D-3D Mutual Information Registration

M.E. Leventon, W.M. Wells III, W.E.L. Grimson
Massachusetts Institute of Technology Artificial Intelligence Laboratory
545 Technology Square, Cambridge, MA 02139
E-MAIL: leventon@ai.mit.edu
HOMEPAGE: http://www.ai.mit.edu/people/leventon

Abstract

We present a method for finding the pose of an object in the world by registering a $3D$ model of the object to multiple images of the object taken from different positions by maximization of mutual information. Using multiple views of the object enables the registration process to converge on the three dimensional pose much more accurately than is possible from using just a single view. Since this method uses mutual information, the model of the object need only contain information about the shape of the object and need not know any details about other surface properties. Furthermore, this method is robust with respect to variations of illumination in the images. The method does not attempt to find any correspondences between pixels in the images, so the images of the object can be obtained from drastically different views and under different lighting conditions.

1 Introduction

Accurately computing the alignment of a $3D$ model of an object to an image of the object is an important problem in many computer vision applications. Many images of the same object can look very different, depending on object pose, lighting conditions, and other objects in the image. Therefore, the problem of computing the registration is a challenging one. One of the advantages of a mutual information approach to registration is that it is robust to many of the unknowns that can occur in an image, including lighting conditions and occlusions [Viola and Wells, 1995]. Viola and Wells used this approach to compute a very accurate registration of a $3D$ model of an object to that object's position in the image plane. However, given that only one image was used to register the object, the registration found was not very accurate along the direction of the optical axis. Their error for registrations of a plastic skull were mostly under $2mm$ in the x and y dimension, but ranged from $5mm$ to almost $15mm$ in the z dimension [Viola and Wells, 1995]. A change in an object's x and y position is very noticeable in an image, while a shift in model position along the optical axis is difficult to see.

However, many applications require more than just an accurate registration in the image frame; an accurate $3D$ pose of the object is often needed. For example, a surgeon might want to register a patient's head to his or her internal CT/MR scan in order to very accurately position the patient for radiation therapy. During neurosurgery, the surgeon might want to point a trackable probe at some position inside the brain, and have the system display the $3D$ position of the probe in the CT/MR scan. Clearly, in these applications, being accurate in two dimensions is not enough. The motivation behind performing a mutual information registration using multiple views is to take advantage of the robustness in illumination variation and occlusion that mutual information offers, while also being able to accurately compute the pose of an object in three dimensions.

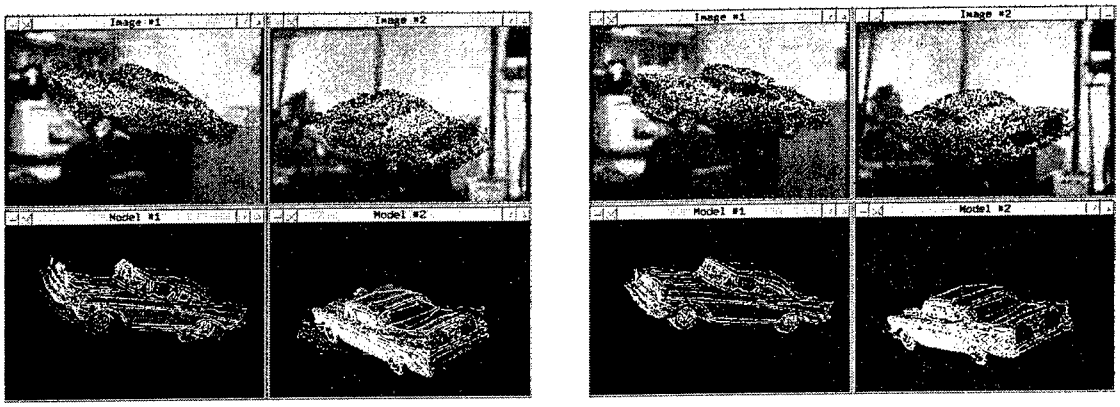


Figure 1: The first figure shows the initial random alignment of a test registration. The top two images show the views from the two cameras with randomly selected model points overlaid in red. The bottom two images show the model transformed by the pose and then projected into both image planes. The error associated with the initial pose is $50.8mm$. The second figure shows the result of using both views to register the model. The error in this final pose is $3.1mm$.

2 Previous Work

Much work has been done on the problem of registering a $3D$ model of an object to the world position of that object. Stereo methods of registration have the potential to improve the alignment along the optical axis, since depth information can be computed. However, stereo is susceptible to difficulty in finding correspondences between pixels in the images.

The difficulty in feature-based image registration lies in the problem of extracting common features between the model and the image. For example, edges extracted from an image can be due to albedo change, surface normal change, or illumination change (i.e. shadowing). The only types of edges that could be extracted from our shape model are edges due to change in surface normal. However, many objects have varying albedo and also shadow themselves, which will lead to many spurious edges in the image.

Fiducial registration involves manually picking corresponding points from the $3D$ model and the object. Accurately localizing these points is often difficult. For neurosurgical applications, Peters [Peters *et al.*, 1996] reports fiducial accuracy about an order magnitude worse than frame-based methods of registration, mainly due to the difficulty in accurately localizing the fiducial markers in both the internal scan and also on the patient.

3 Mutual Information Registration

In this section we review the basic method of alignment by maximization of mutual information, which has been described previously, [Viola and Wells, 1995] [Viola, 1995] [Wells *et al.*, 1995].

We seek an estimate of the transformation $\hat{T}$ that aligns the model u and image v by maximizing their mutual information over the transformations T,

$$\hat{T} = \arg\max_{T} I(u(x),\ v(T(x))) \quad .$$

Here x is a random variable that ranges over visible surface patches in the model.

Mutual information is defined in terms of entropy in the following way:

$$I(u(x),\ v(T(x))) \equiv \qquad (1)$$
$$H(u(x)) + H(v(T(x))) - H(u(x), v(T(x)))$$

. $H(\cdot)$ is the entropy of a random variable, and is defined as $H(x) \equiv -\int p(x) \ln p(x) dx$. The joint entropy of two random variables x and y is $H(x,y) \equiv -\int p(x,y) \ln p(x,y) dx dy$. Entropy can be interpreted as a measure of uncertainty, variability, or complexity.

Information has three components. The first term on the right in Equation 1 is the entropy in the model. It is not a function of T. The second

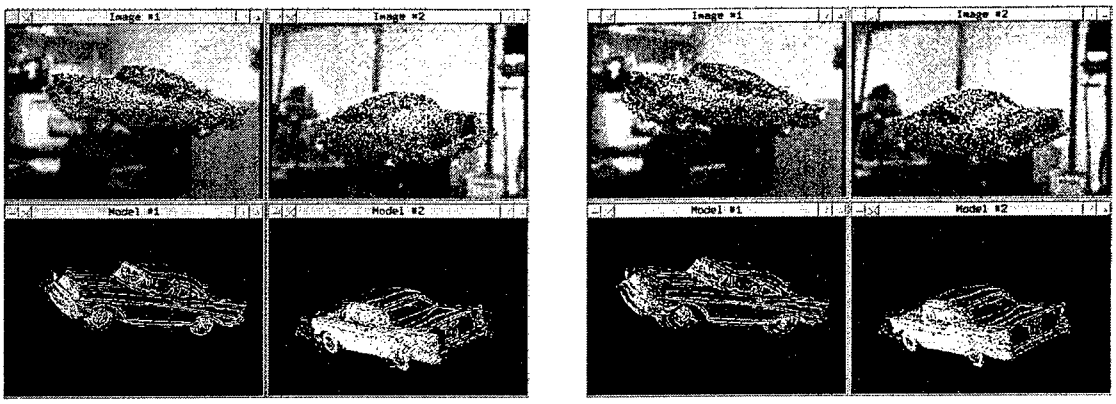


Figure 2: The first figure shows the result of just using the *first* image to register, and the second figure shows the result of only using the *second* image to register. Notice that in both cases, the registration is good in the image plane of the view that was used but is off in the other view.

term is the entropy of the part of the image into which the model projects. It encourages transformations that project u into complex parts of v. The third term, the (negative) joint entropy of u and v, takes on large values if u and v are functionally related. It encourages transformations where u explains v well. Together the last two terms identify transformations that find complexity and explain it well. This is the essence of mutual information.

In [Viola and Wells, 1995] and [Viola, 1995] a stochastic gradient descent method was used to seek local maxima of the mutual information criterion, and [Wells *et al.*, 1995] described a gradient method that uses histograms to approximate entropies and their derivatives. The latter method was used in the work reported here.

4 Multiple View Registration

The goal of the multiple view 2D-3D mutual information registration approach is to find the pose of the model that best describes all the images of the object. The algorithm is very similar to that which is described in [Viola and Wells, 1995]. To apply the mutual information registration technique to multiple views, we perform a single-view registration "step" for each view in turn.

The algorithm requires a point/normal model M of the object, and n images of that object $\{I_1, \ldots, I_n\}$. Additionally, the relative poses (positions and orientations) of the n cameras must be known. Let $T_j \in \{T_1, \ldots, T_n\}$ be the transformation that takes a point in world coordinates into Camera j's coordinates. (Assume, for simplicity, that T_1 is the identity, so world coordinates are the same as Camera 1 coordinates.) Finally, the algorithm requires an initial pose P_0 from which to perform the gradient ascent.

At each iteration, for each view, the algorithm updates the pose in the direction of the gradient of mutual information for that view.

$P \leftarrow P_0$
For each iteration i, $(i = 1, 2, \ldots)$
  For each view j, $(j = 1, \ldots, n)$
    Define:
      $P_j \leftarrow PM$
      $M_j \leftarrow T_j P_j$
    Compute:
      $\Delta P_j \leftarrow \nabla MI(M_j, I_j)$
    Let:
      $D = translation(P_j)$
      $R = quaternion_rotation(P_j)$
      $d = \lambda_d \times translation(\Delta P_j)$
      $r = scale_rotation$
        $(quaternion_rotation(\Delta P_j), \lambda_r)$
    Update:
      $P'_j(\cdot) \leftarrow r(R(\cdot)) + D + d$
      $P \leftarrow T_j^{-1} P'_j$

The functions *translation*(P) and *quaternion_rotation*(P) extract the translational and rota-

tional components of the pose P respectively. A rotation r can be represented as a rotation angle θ about some unit vector v (so $r = \langle \theta, v \rangle$). The function *scale_rotation* scales the rotation angle θ such that $scale_rotation(\langle \theta, v \rangle, \lambda_r) = \langle \lambda_r \theta, v \rangle$. Note that in the first update step, P'_j is set to the result of composing P_j with a scaled ΔP_j

5 Results

In this section, we present the results of running multiple registration experiments using two views of a model car. A $3D$ point-normal model of the car was derived from a computed tomography (CT) scan. Two cameras were placed approximately $1.5m$ apart aimed at the model car that is $0.5m$ in length and positioned about $1.0m$ away from the two cameras. The images of the car taken from the two cameras are shown in figures 1. A "correct" pose was determined by manually aligning the $3D$ model in both image frames. This pose will be used as the ground truth for the registration experiments.

In order to evaluate a pose that is returned by the registration algorithm, it is necessary to define a distance or error metric for poses. The error metric we used is defined as follows:

$$E_{3D}(P|P^*, M) = \max_{q \in M} ||Pq - P^*q||$$

The error in pose, given the "correct" pose P^* and the model M, is the maximum distance between corresponding model points under the two transformations.

5.1 Results on One Example Trial

Starting with a initial random pose with an error of $50.8mm$, after 200 iterations of the algorithm, using both views, the final pose error is $3.1mm$. Figure 1 shows the initial pose and the final pose. As a comparative measure, the algorithm was run twice more, once using only the first view, and a second time using only the second view, again for 200 iterations starting from the same initial pose. The pose errors for these single-view registrations were $17.3mm$ and $26.8mm$ respectively. Figure 2 shows the results of these registrations. Notice that for both registrations, the algorithm did a good job of locking down the registration in the x and y direction of the view it processed, but did not register the model very accurately in the z direction, or the direction of the optical axis. In both cases, this error in registration is noticeable in the other view of the car that was not used in the registration.

5.2 Results Over Many Trials

The two-view mutual information registration algorithm was run on the same car images (in figure 1) with 400 random starting model poses. Each random initial pose was with in $\pm 15mm$ and $\pm 10°$ in each dimension of the "right" pose. For each random pose, the algorithm was run for 200 iterations, once using only the first view, once using only the second view, and once using both views. The graph in Figure 3 shows the results from these 400 trials, sorted by error. As seen in the graph, the two view registration process aligned the model within $3.5mm$ of the correct pose slightly over 80% of the time. When the algorithm had only one view to work with, the registration error was significantly greater, and from the graph, it is even difficult to see when it converged near the correct solution and when it found an incorrect pose.

However, one of the first observations and the main motivation for using multiple views was that a single view has very limited depth information, and *any* registration algorithm would have difficulty accurately registering an object along the optical axis. Thus, comparing the single-view and two-view algorithms in this way — by using three dimensional distance — is not really a fair comparison. Therefore, we define the following two dimensional error metric:

$$E_{2D}(P|P^*, M) = \max_{q \in M} ||F_P(Pq) - F_P(P^*q)||$$

where F_P takes a point in the $3D$ coordinates of a camera and projects it into $2D$ image coordinates. E_{2D}, unlike E_{3D}, only considers the error in the image plane, and ignores any error along the optical axis. Thus, E_{2D} would seem to be a more "fair" error measurement to use when comparing a multiple-view registration to a single-view registration.

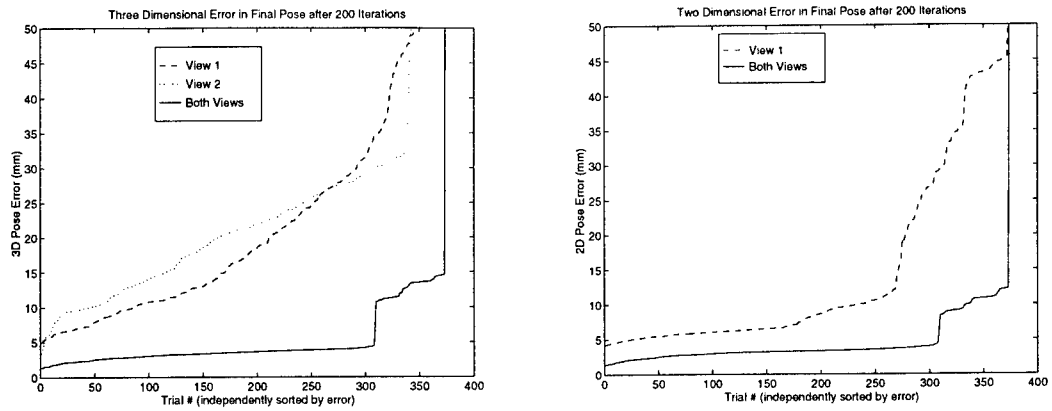


Figure 3: The first graph illustrates the $3D$ error in pose after 200 iterations over the 400 registrations with random initial poses. The second graph illustrates the $2D$ pose error. In this graph, any error along the optical axis is ignored.

Figure 3 shows the results of the 400 trials using the two dimensional error metric. Notice that now it is much more obvious when the single-view algorithm converged near the correct solution. However, the two-view approach still performs significantly better in a few different ways. First, the pose error of the two-view algorithm is still much less that of the single-view algorithm, usually by a factor of two. This implies that using two views not only improves the registration along the optical axis, but also yields a better registration in the image plane.

In addition to returning more accurate registrations, the two-view method also seems to have a much larger region of convergence. Of the 400 trials, the two-view approach registered well 80% of the time, registered reasonably about 15% of the time, and diverged the remaining 5%. The single view method registered well only about 50% of the time, registered reasonably about 25% of the time, and diverged about 25% of the time. Thus, using two views seems to drastically improve the registration of the model to the position of the $3D$ object in the world.

One disadvantage of using multiple views is that it can be slower by a factor of n for n views. Generally, though, the registration actually converges much faster (and is more likely to converge), so the slowdown may not be significant. Another disadvantage of using n views is that it requires the calibration of n cameras. In some applications, it might be difficult to precisely calibrate the n cameras.

References

[Grimson *et al.*, 1996] W.E.L. Grimson, G.J. Ettinger, S.J. White, T. Lozano-Pérez, W.M. Wells III, and R. Kikinis. "An Automatic Registration Method for Frameless Stereotaxy, Image Guided Surgery, and Enhanced Reality Visualization". *IEEE TMI*, **15**(2):129–140, April 1996.

[Horn, 1987] B. Horn. "Closed-form Solution of Absolute Orientation Using Unit Quaternions". *JOSA A*, **4**:629–642, April 1987.

[Peters *et al.*, 1996] T. Peters, B. Davey, P. Munger, R. Comeau, A. Evans, A. Olivier. "Three-Dimensional Multimodal Image-Guidance for Neurosurgery". *IEEE TMI*, **15**(2):121–128, April 1996.

[Viola and Wells, 1995] P.A. Viola, W.M. Wells III. "Alignment by Maximization of Mutual Information". In *International Conference on Computer Vision*, June 1995.

[Viola, 1995] P.A. Viola. PhD thesis. MIT Department Electrical Engineering and Computer Science, Cambridge, Mass., 1995.

[Wells *et al.*, 1995] W. Wells III, M. Halle, R. Kikinis, P. Viola. "Alignment and Tracking using Graphics Hardware". In *Proceedings of the Image Understanding Workshop*, Feb. 1996.

Minimizing Algebraic Error *

Richard I. Hartley,
G.E. Corporate Research and Development
1 Research Circle
Niskayuna, NY 12309

Abstract

This paper gives a widely applicable technique for solving many of the parameter estimation problems encountered in geometric computer vision. A commonly used approach in such parameter minimization is to minimize an algebraic error function instead of a possibly preferable geometric error function. It is claimed in this paper, however, that minimizing algebraic error will usually give excellent results, and in fact the main problem with most algorithms minimizing algebraic distance is that they do not take account of mathematical constraints that should be imposed on the quantity being estimated. This paper gives an efficient method of minimizing algebraic distance while taking account of the constraints. This provides new algorithms for the problems of resectioning a pinhole camera, computing the fundamental matrix, and computing the tri-focal tensor.

1 Introduction

For many problems related to camera calibration and scene reconstruction, linear algorithms are known for solving for the entity required. In the sort of problem that will be addressed in this paper, a set of data (such as point correspondences) is used to construct a set of linear equations, and solution of these equations provides an estimate of the entity being computed. As examples of such problems we have :

1. The DLT algorithm for computing a camera matrix given a set of points in space, and corresponding points in the image. Provided at least 6 correspondences are give (more precisely $5\frac{1}{2}$ correspondences), one can solve for the camera matrix.
2. Computation of the Fundamental Matrix. From 8 point correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i$ between two images one can construct the fundamental matrix using equations $\mathbf{u}'^{\top}_i F \mathbf{u}_i = 0$.
3. Computation of the trifocal tensor given a set of feature correspondences across three views.

In these three examples, and many others, a linear algorithm exists. However, the linear algorithm will lead to a solution that does not satisfy certain constraints that the estimated quantity must satisfy. In the cases considered here, the constraints are

1. The *skew* parameter of a camera matrix estimated using the DLT method will not generally be zero. This constraint, meaning the pixels are rectangular, should be enforced in cases where it is known to hold.
2. The fundamamental matrix must satisfy a constraint $\det F = 0$.
3. The trifocal tensor must satisfy 8 non-linear constraints. The form of these constraints is not easily determined, but it is essential to constrain the tensor to correspond to a valid set of camera matrices.

These constraints are not in general linear constraints, and in general, it will be necessary to resort to iterative techniques to enforce them. Since iterative techniques are slow and potentially unstable, it is important to use them sparingly. Further, the smaller the dimension of the minimization problem, the faster and generally more stable the solution will be. In this paper an iterative algorithm is used to solve the three problems posed above. In each case the algorithms are based on a common technique of data reduction, whereby the input data is condensed into a *reduced measurement matrix*. The size of the iteration problem is then independent on the size of the input set. In the case of estimation of the fundamental matrix, only three homogeneous parameters are used to parametrize the minimization problem, whereas for the trifocal tensor, just six parameters are used.

The problem of camera calibration solved using the DLT algorithm will be treated first. It will be used to illustrate the techniques that apply to the other problems.

*This work was sponsored by DARPA contract F33615-94-C-1549, monitored by Wright Patterson Airforce Base, Dayton, OH. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the United States Government, or General Electric.

2 Computing the Camera Matrix

We consider a set of point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{u}_i$ between 3D points $\mathbf{x}_i$ and image points $\mathbf{u}_i$. Our problem is to compute a 3×4 matrix P such that $P\mathbf{x}_i = \mathbf{u}_i$ for each i.

2.1 The Direct Linear Transformation (DLT) algorithm

We begin with a simple linear algorithm for determining P given a set of 3D to 2D point correspondences, $\mathbf{x}_i \leftrightarrow \mathbf{u}_i$. The correspondence is given by the equation $\mathbf{u}_i = P\mathbf{x}_i$. Note that this is an equation involving homogeneous vectors, thus $\mathbf{u}_i$ and $P\mathbf{x}_i$ may differ by a non-zero scale factor. One may, however write the equation in terms of the vector cross product as $\mathbf{u}_i \times P\mathbf{x}_i = 0$.

If the j-th row of the matrix P is denoted by $\mathbf{p}^{j\top}$, then we may write $P\mathbf{x}_i = (\mathbf{p}^{1\top}\mathbf{x}_i, \mathbf{p}^{2\top}\mathbf{x}_i, \mathbf{p}^{3\top}\mathbf{x}_i)^\top$. Writing $\mathbf{u}_i = (u_i, v_i, w_i)^\top$, The cross product may then be given explicitly as

$$\mathbf{u}_i \times P\mathbf{x}_i = \begin{pmatrix} v_i\mathbf{p}^{3\top}\mathbf{x}_i - w_i\mathbf{p}^{2\top}\mathbf{x}_i \\ w_i\mathbf{p}^{1\top}\mathbf{x}_i - u_i\mathbf{p}^{3\top}\mathbf{x}_i \\ u_i\mathbf{p}^{2\top}\mathbf{x}_i - v_i\mathbf{p}^{1\top}\mathbf{x}_i \end{pmatrix} .$$

Since $\mathbf{p}^{j\top}\mathbf{x}_i = \mathbf{x}_i{}^\top\mathbf{p}^j$ for $j = 1, \ldots, 3$, this gives a set of three equations, in the entries of P, which may be written in the form

$$\begin{bmatrix} 0 & -w_i\mathbf{x}_i{}^\top & v_i\mathbf{x}_i{}^\top \\ w_i\mathbf{x}_i{}^\top & 0 & -u_i\mathbf{x}_i{}^\top \\ -v_i\mathbf{x}_i{}^\top & u_i\mathbf{x}_i{}^\top & 0 \end{bmatrix} \begin{pmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{pmatrix} = \mathbf{0} . \quad (1)$$

Note that $(\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3)^\top$ which appears in (1) is a 12-vector made up of the entries of the matrix P. Although there are three equations, only two of them are linearly independent. Thus each point correspondence gives two equations in the entries of P. One may choose to omit the third equation, or else include all three equations, which may sometimes give a better conditioned set of equations. In future, we will assume that only the first two equations are used, namely

$$\begin{bmatrix} 0 & -w_i\mathbf{x}_i{}^\top & v_i\mathbf{x}_i{}^\top \\ w_i\mathbf{x}_i{}^\top & 0 & -u_i\mathbf{x}_i{}^\top \end{bmatrix} \begin{pmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{pmatrix} = \mathbf{0} . \quad (2)$$

Solving the Equations. The equations (2) may be denoted by $M_i\mathbf{p} = \mathbf{0}$. where the vector $\mathbf{p}$ is a 12-vector, corresponding to the 12 entries of P. The set of all equations derived from several point correspondences may be written $M\mathbf{p} = \mathbf{0}$ where M is the matrix of equation coefficients. This matrix M will be called the *measurement matrix*. The obvious solution $\mathbf{p} = 0$ is of no interest to us, so we seek a non-zero solution $\mathbf{p}$.

2.2 Scaling

One of the most important things to do in implementing an algorithm of this sort is to prenormalize the data. This type of data normalization was discussed in the paper [2]. Without this normalization, all these algorithms are guaranteed to perform extremely poorly.

Data normalization is designed to improve the conditioning of the measurement matrix M. The appropriate scaling is to translate all data points so that their centroid is at the origin. Then the data should be scaled so that the average distance of any data point from the origin is equal to $\sqrt{2}$ for image points and $\sqrt{3}$ for 3D points. The algorithms are then carried out with the normalized data, and final transformations are applied to the result to compensate for the normalizing transforms.

2.3 Algebraic Error

In the presence of noise, one can not expect to obtain an exact solution to an overconstrained set of equations of the form $M\mathbf{p} = \mathbf{0}$ such as those that arise in the DLT method.

The DLT algorithm instead finds the unit-norm vector $\mathbf{p}$ that minimizes $||M\mathbf{p}||$. The vector $\epsilon = M\mathbf{p}$ is the error vector and it is this error vector that is minimized. The solution is the unit singular vector corresponding to the smallest singular value of M.

Define a vector $(\hat{u}_i, \hat{v}_i, \hat{w}_i)^\top = \hat{\mathbf{u}}_i = P\mathbf{x}$. Using this notation, we may write

$$M_i\mathbf{p} = \epsilon_i = \begin{pmatrix} v_i\hat{w}_i - w_i\hat{v}_i \\ w_i\hat{u}_i - u_i\hat{w}_i \end{pmatrix} = \mathbf{0} . \quad (3)$$

This vector is the *algebraic error vector* associated with the point correspondence $\mathbf{u}_i \leftrightarrow \mathbf{x}_i$ and the camera mapping P. Thus,

$$d_{\text{alg}}(\mathbf{u}_i, \hat{\mathbf{u}}_i)^2 = (v_i\hat{w}_i - w_i\hat{v}_i)^2 + (w_i\hat{u}_i - u_i\hat{w}_i)^2 . \quad (4)$$

Given several point correspondences, the quantity $\epsilon = M\mathbf{p}$ is the algebraic error vector for the complete set, and one sees that

$$\sum_i d_{\text{alg}}(\mathbf{u}_i, \hat{\mathbf{u}}_i)^2 = ||M\mathbf{p}||^2 = ||\epsilon||^2 \quad (5)$$

The main lesson that we want to keep from this discussion is :

Proposition 1. *Given any set of 3D to image correspondences $\mathbf{u}_i \leftrightarrow \mathbf{x}_i$, let M be the measurement matrix as in (2). For any camera matrix P the vector $M\mathbf{p}$ is the algebraic error vector, where $\mathbf{p}$ is the vector of entries of P.*

2.4 Geometric Distance

Under the assumption that measurement error is confined to image measurements, and an assumption of a gaussian error model for the measurement of 2D image coordinates, the optimal estimate for the camera matrix P is the one that minimizes the error function

$$\sum_i d(\mathbf{u}_i, \hat{\mathbf{u}}_i)^2 \quad (6)$$

where $d(\cdot, \cdot)$ represents Euclidean distance in the image. The quantity $d(\mathbf{u}_i, \hat{\mathbf{u}}_i)$ is known as the *geometric distance* between $\mathbf{u}_i$ and $\hat{\mathbf{u}}_i$. Thus the error to be minimized is the sum of squares of geometric distances between measured and projected points.

For points $\mathbf{u}_i = (u_i, v_i, w_i)^\top$ and $\hat{\mathbf{u}}_i = (\hat{u}_i, \hat{v}_i, \hat{w}_i)^\top$, the geometric distance is

$$\begin{aligned} d(\mathbf{u}_i, \hat{\mathbf{u}}_i) &= \left((u_i/w_i - \hat{u}_i/\hat{w}_i)^2 + (v_i/w_i - \hat{u}_i/\hat{w}_i)^2\right)^{1/2} \\ &= d_{\text{alg}}(\mathbf{u}_i, \hat{\mathbf{u}}_i)/w_i w_i' \end{aligned} \quad (7)$$

Thus, geometric distance is related to, but not quite the same as algebraic distance. Nevertheless, it will turn out that minimizing algebraic distance gives very good results in general.

2.5 The Reduced Measurement Matrix

Let $\mathbf{u}_i \leftrightarrow \mathbf{x}_i$ be a set of correspondences, and let M be the corresponding measurement matrix. Let P be any camera matrix, and let $\mathbf{p}$ be the vector containing its entries. The algebraic error vector corresponding to P is $M\mathbf{p}$, and its norm satisfies $||M\mathbf{p}||^2 = \mathbf{p}^\top M^\top M\mathbf{p}$.

In general, the matrix M may have a very large number of rows. It is possible to replace M by a square matrix $\hat{M}$ such that $||M\mathbf{p}|| = ||\hat{M}\mathbf{p}||$ for any vector $\mathbf{p}$. Such a matrix $\hat{M}$ is called a *reduced measurement matrix*. One way to do this is using the Singular Value Decomposition (SVD). Let $M = UDV^\top$ be the SVD of M, and define $\hat{M} = DV^\top$. Then

$$M^\top M = (VDU^\top)(UDV^\top) = (VD)(DV^\top) = \hat{M}^\top \hat{M}$$

as required. Another way of obtaining $\hat{M}$ is to use the QR decomposition $M = Q\hat{M}$, where Q has orthogonal columns and $\hat{M}$ is upper-triangular and square. This shows the following result.

Theorem 2. *Let $\mathbf{u}_i \leftrightarrow \mathbf{x}_i$ be a set of n world to image correspondences. Let M be the measurement matrix derived from the point correspondences. Let $\hat{M}$ be a reduced measurement matrix. Then, for any 3D to 2D projective transform P and corresponding 3-vector $\mathbf{p}$, one has*

$$\sum_i d_{\text{alg}}(\mathbf{u}_i, P\mathbf{x}_i)^2 = ||\hat{M}\mathbf{p}||^2$$

In this way, all the information we need to keep about the set of matched points $\mathbf{u}_i \leftrightarrow \mathbf{x}_i$ is contained in the single 12×12 matrix $\hat{M}$. If we wish to minimize algebraic error as P varies over some restricted set of transforms, then this is equivalent to minimizing the norm of the 12-vector $||\hat{M}\mathbf{p}||$.

2.6 Restricted Camera Mappings

The camera mapping expressed by a general 3D projective transformation is in some respects too general. A non-singular 3×4 matrix P with center at a finite point may be decomposed as $P = K[R \mid -R\mathbf{t}]$ where R is a 3×3 rotation matrix and

$$K = \begin{bmatrix} \alpha_u & s & u_0 \\ & \alpha_v & v_0 \\ & & 1 \end{bmatrix} . \qquad (8)$$

The non-zero entries of K are geometrically meaningful quantities, the internal calibration parameters of P. A common assumption is that $s = 0$, while for a true pinhole camera, $\alpha_u = \alpha_v$.

Given a set of world to image correspondences, one may wish to find a matrix P that minimizes algebraic error, subject to a set of constraints on P. Usually, this will require an iterative solution. For instance, suppose we wish to enforce the constraints $s = 0$ and $\alpha_u = \alpha_v$. One can parametrize the camera matrix using the remaining 9 parameters (p_u, p_v, α plus 6 parameters representing the orientation R and location $\mathbf{t}$ of the camera). Let this set of parameters be denoted collectively by $\mathbf{q}$. Then, one has a map $\mathbf{p} = g(\mathbf{q})$, where $\mathbf{p}$ is as before the vector of entries of the matrix P. According to Theorem 2, minimizing algebraic error over all point matches is equivalent to minimizing $||Mg(\mathbf{q})||$. Note that the mapping $\mathbf{q} \mapsto Mg(\mathbf{q})$ is a mapping from R^9 to R^{12}. This is a simple parameter-minimization problem that may be solved using the Levenberg-Marquardt method. The important point to note is the following :

> Given a set of n world-to-image correspondences, $\mathbf{x}_i \leftrightarrow \mathbf{u}_i$, the problem of finding a constrained camera matrix P that minimizes the sum of algebraic distances $\sum_i d_{\text{alg}}(\mathbf{u}_i, P\mathbf{x}_i)^2$ reduces to the minimization of a function $R^9 \rightarrow R^{12}$, independent of the number n of correspondences.

If this problem is solved using the Levenberg-Marquardt (LM) method, then an initial estimate of the parameters may be obtained by decomposing a camera matrix P found using the DLT algorithm. A central step in the LM method is the computation of the derivative matrix (Jacobian matrix) of the function being minimized, in this case $Mg(\mathbf{q})$. Note that $\partial Mg/\partial \mathbf{q} = M\partial g/\partial \mathbf{q}$. Thus, computation of the Jacobian reduces to computation of the Jacobian matrix of g, and subsequent multiplication by M.

Minimization of $||Mg(\mathbf{q})||$ takes place over all values of the parameters $\mathbf{q}$. Note, however, that if $P = K[R \mid -R\mathbf{t}]$ with K as in (8) then P satisfies the condition $p_{31}^2 + p_{32}^2 + p_3^2 = 1$, since these entries are the same as the last row of the rotation matrix R. Thus, minimizing $Mg(\mathbf{q})$ will lead to a matrix P satisfying the constraints $s = 0$ and $k_u = k_v$ and scaled such that $p_{31}^2+p_{32}^2+p_3^2 = 1$, and which in addition minimizes the algebraic error for all point correspondences.

3 Computation of the Fundamental Matrix

We now turn to the computation of the Fundamental Matrix. It will turn out that very similar methods apply to its computation as were used in the DLT algorithm.

Given a set of correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i$ between two images, the fundamental matrix is defined by the relation $\mathbf{u}_i'^T F\mathbf{u}_i = 0$ for all i. In the presence of noise, this relation will not hold precisely, and so one seeks a least-squares solution. Note that the equation $\mathbf{u}'_i F\mathbf{u}_i = 0$ is linear in the entries of F. From 8 or more point matches, one may solve for the entries of F by finding the least-squares solution to a set of linear equations ([2]). Let the set of equations be denoted by $M\mathbf{f} = 0$. The vector $M\mathbf{f}$ has components equal to $\mathbf{u}_i'^T F\mathbf{u}_i$, and $||M\mathbf{f}||^2 = \sum_i (\mathbf{u}'_i F\mathbf{u}_i)^2$. Thus, in this case, as before with the DLT algorithm , $M\mathbf{f}$ represents the algebraic error vector. Matrix F is found my minimizing $||M\mathbf{f}||$ subject to $||\mathbf{f}|| = 1$.

The fundamental matrix F must, however satisfy a constraint $\det F = 0$, and this constraint will not gen-

erally be satisfied by the matrix F found by this linear algorithm. One would therefore like to minimize the algebraic error $||M\hat{\mathbf{f}}||$ over all vectors $\hat{\mathbf{f}}$ corresponding to singular matrices $\hat{F}$.

In [2], the matrix $\hat{F}$ was taken to be the closest singular matrix to F under Frobenius norm, where F is the linear solution. This is not an especially good way of proceeding, since it weights errors in each of the entries of F equally. A preferable method is to proceed as with the DLT. One parametrizes the matrix $\hat{F}$ by a set of parameters $\mathbf{q}$ in a way so as to ensure it is singular. Then letting $\hat{f} = g(\mathbf{q})$, one uses an iterative algorithm to minimize $||Mg(\mathbf{q})||$. This is the general scheme which will be followed, but there are details to be filled out, and a new twist will arise, which allows a parametrization with only three parameters.

3.1 Parametrization of the Fundamental Matrix

Consider the fundamental matrix F, which can be written as a product $F = Q[\mathbf{e}]_\times$ where Q is a non-singular matrix, and $\mathbf{e}$ is the epipole in the first image.

Suppose we wish to compute the fundamental matrix F of the form $F = Q[\mathbf{e}]_\times$ that minimizes the algebraic error $||M\mathbf{f}||$ subject to the condition $||\mathbf{f}|| = 1$. The vector $\mathbf{f}$ is the 9-vector containing the entries of F. It has been seen that the 8-point algorithm finds such an $\mathbf{f}$, without the condition that $F = Q[\mathbf{e}]_\times$. We now wish to enforce that condition.

Let us assume for now that the epipole $\mathbf{e}$ is known. Later we will let $\mathbf{e}$ vary, but for now it is fixed. The equation $F = Q[\mathbf{e}]_\times$ can be written in terms of the vectors $\mathbf{f}$ and $\mathbf{q}$ comprising the entries of F and Q as an equation $\mathbf{f} = E\mathbf{q}$ where E is a 9×9 matrix. Supposing that $\mathbf{f}$ and $\mathbf{q}$ contain the entries of the corresponding matrices in row-major order, then it can be verified that E has the form

$$E = \begin{bmatrix} [\mathbf{e}]_\times & & \\ & [\mathbf{e}]_\times & \\ & & [\mathbf{e}]_\times \end{bmatrix} . \qquad (9)$$

Now, our minimization problem is : minimize $||ME\mathbf{q}||$ subject to the condition $||E\mathbf{q}|| = 1$.[1] This problem is solved as follows. Let the Singular Value Decomposition of E be $E = UDV^\top$. It is easily seen that the matrix E has rank 6, since each of the diagonal blocks has rank 2. It follows that D has 6 non-zero diagonal entries. Let U' be the 9×6 matrix consisting of the first 6 columns of U, and let V' consist of the first 6 columns of V and let D' be the top-left 6×6 minor of D, containing the non-zero diagonal entries. The minimization problem them becomes : minimize $||MU'D'V'^\top\mathbf{q}||$ subject to $||U'D'V'^\top\mathbf{q}|| = 1$. This last condition is equivalent to $||D'V'^\top\mathbf{q}|| = 1$, since U' has orthogonal columns. Now writing $\mathbf{q}' = D'V'^\top\mathbf{q}$, the problem becomes : minimize $||MU'\mathbf{q}'||$ subject to $||\mathbf{q}'|| = 1$, which is our standard minimization problem. The solution $\mathbf{q}'$ is the singular vector corresponding to the smallest singular value of MU'. Subsequently, we can compute $\mathbf{f} = E\mathbf{q} = U'D'V'^\top\mathbf{q} = U'\mathbf{q}'$, and the algebraic error is $M\mathbf{f} = MU'\mathbf{q}'$.

The complete algorithm is :

Algorithm

1. Given the epipole $\mathbf{e}$, find the fundamental matrix F of the form $F = Q[\mathbf{e}]_\times$ that minimizes the algebraic error $||M\mathbf{f}||$ subject to $||\mathbf{f}|| = 1$.

Solution :

1. Compute the SVD $E = UDV^\top$, where E is given in (9).
2. Let U' be the matrix comprising the first 6 columns of U
3. Find the unit vector $\mathbf{q}'$ that minimizes $||MU'\mathbf{q}'||$.
4. The required matrix F corresponds to the vector $\mathbf{f} = U'\mathbf{q}'$, and the minimum algebraic error is $M\mathbf{f}$.

3.2 Iterative Estimation

The algorithm of the last section gives a way of computing an algebraic error vector $M\mathbf{f}$ given a value for the epipole $\mathbf{e}$. This mapping $\mathbf{e} \mapsto M\mathbf{f}$ is a map from $R^3 to R^9$. Note that the value of $M\mathbf{f}$ is unaffected by scaling $\mathbf{e}$. Starting from a value of $\mathbf{e}$ derived as the generator of the right null-space of an initial estimate of F, one may iterate to find the final F that minimizes algebraic error. The initial estimate of F may be obtained from the 8-point algorithm, or any other simple algorithm.

Note the advantage of this method of computing F is that the iterative part of the algorithm consists of a very small parameter minimzation problem, involving the estimation of only three parameters. Despite this, the algorithm finds the fundamental matrix that minimizes the algebraic error for all matched points. The matched points themselves do not come into the final iterative estimation.

Simplifying the computation Because of the simple form of the matrix E, it is easy to compute its SVD without having to resort to a full SVD algorithm. This may be important in the iterative algorithm to achieve maximum speed, since this SVD is computed repeatedly during the minimization. As seen in (9),the matrix E has a diagonal block structure consisting of three blocks $[\mathbf{e}]_\times$. The SVD consequently has a corresponding block-structure. Specifically, if $[\mathbf{e}]_\times = \hat{U}\hat{D}\hat{V}'$, then the SVD of $E = \mathrm{diag}([\mathbf{e}]_\times, [\mathbf{e}]_\times, [\mathbf{e}]_\times)$ is $E = UDV^\top$ where $U = \mathrm{diag}(\hat{U}, \hat{U}, \hat{U})$, and similarly for D and V.

The SVD of $[\mathbf{e}]_\times$ itself can be computed easily as follows. Suppose that $\hat{U}$ is an orthogonal matrix such that $\mathbf{e}\hat{U} = (0, 0, 1)$. Such a matrix $\hat{U}$ is a Householder transformation and is easily computed ([1]). Then one sees that $[\mathbf{e}]_\times = \pm\hat{U}Z\hat{U}^\top = \pm\hat{U}\,\mathrm{diag}(1, 1, 0)\hat{Z}\hat{U}^\top = \pm\hat{U}\hat{D}\hat{V}^\top$ where

$$Z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad ; \quad \hat{Z} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This is easily verified by observing that both $[\mathbf{e}]_\times$ and $\pm\hat{U}Z\hat{U}^\top$ are skew-symmetric matrices with the same

[1] It does not do to minimize $||ME\mathbf{q}||$ subject to the condition $||\mathbf{q}|| = 1$, since a solution to this occurs when $\mathbf{q}$ is a unit vector in the right null-space of E. In this case, $E\mathbf{q} = 0$, and hence $||ME\mathbf{q}|| = 0$.

null-space, generated by $\mathbf{e}$ in each case. We are interested in $\hat{U}'$ consisting of the first two columns of $\hat{U}$. Turning now to the SVD of $E = \mathrm{diag}([\mathbf{e}]_\times, [\mathbf{e}]_\times, [\mathbf{e}]_\times)$, we see that $U' = \mathrm{diag}(\hat{U}', \hat{U}', \hat{U}')$. If we partition the 9×9 matrix M into blocks $M = [M_1, M_2, M_3]$ where each M_i has 3 columns, then one computes that $MU' = [M_1\hat{U}', M_2\hat{U}', M_3\hat{U}']$. Thus, the computation of MU' required in Algorithm 1 has two simple steps

1. Compute the 3×3 Householder matrix $\hat{U}$ such that $\mathbf{e}^\mathsf{T}\hat{U} = (0, 0, 1)$, and let $\hat{U}'$ comprise its first two columns.
2. Set $MU' = [M_1\hat{U}', M_2\hat{U}', M_3\hat{U}']$.

3.3 Experimental Evaluation of the Algorithm

A set of experiments were carried out similar to those in [2]. One image from each pair of images used is shown in Fig 1. These images contain a wide variation of measurement noise and placement of the epipoles. For each pair of images, a number n of matched points were chosen and the fundamental matrix was computed. The fundamental matrix computed was shown evaluated against the full set of all matched points, and the residual error was computed. This experiment was done 100 times for each value of n and each pair of images, and the average residual error was plotted against n. This gives an idea of how the different algorithms behave as the number of points is increased.

The results of these experiments are shown and explained in Fig 2. They show that minimizing algebraic error gives essentially indistinguishable results from minimizing the geometric error, but both perform better than the linear normalized 8-point algorithm ([2]).

4 Computation of the Trifocal Tensor

The trifocal tensor ([5, 3]), relates the coordinates of points or lines seen in three views in a similar way to that in which the fundamental matrix relates points in two views.

The basic formula relates a point $\mathbf{u}$ in one image and a pair of lines $\boldsymbol{\lambda}'$ and $\boldsymbol{\lambda}''$ in the other two images. Provided there is a point $\mathbf{x}$ in space that maps to $\mathbf{u}$ in the first image and a point on the lines $\boldsymbol{\lambda}'$ and $\boldsymbol{\lambda}''$ in the other two images, the following identity is satisfied :

$$u^i \lambda'_j \lambda''_k T_i^{jk} = 0 \ . \tag{10}$$

Here we are using tensor notation, in which a repeated index appearing in covariant (lower) and contravariant (upper) positions implies summation over the range of indices (namely, $1, \ldots, 3$).

This equation may be used to generate equations given either point or line correspondences across three images. In the case of a line correspondence, $\boldsymbol{\lambda} \leftrightarrow \boldsymbol{\lambda}' \leftrightarrow \boldsymbol{\lambda}''$ one selects two points $\mathbf{u}_0$ and $\mathbf{u}_1$ on the line $\boldsymbol{\lambda}$, and for each of these points one obtains an equation of the form (10). In the case of a point correspondence $\mathbf{u} \leftrightarrow \mathbf{u}' \leftrightarrow \mathbf{u}''$ one selects any lines $\boldsymbol{\lambda}'$ and $\boldsymbol{\lambda}''$ passing through $\mathbf{u}'$ and $\mathbf{u}''$ respectively. Then (10) provides one equation. Four equations are generated from a single 3-view point correspondence by choosing two lines through each of $\mathbf{u}'$ and $\mathbf{u}''$, each pair of lines giving rise to a single equation.

The equations (10) give rise to a set of equations of the form $M\mathbf{t} = 0$ in the 27 entries of the trifocal tensor. From these equations, one may solve for the entries of the tensor. As before, for any tensor T_i^{jk} the value of $M\mathbf{t}$ is the algebraic error vector associated with the input data.

Consider the analogy with the 8-point algorithm for computing the fundamental matrix in the two-view case. The fundamental matrix has a constraint $\det F = 0$ that is not in general precisely satisfied by the solution found from linear algorithm. In the case of the trifocal tensor, there are 27 entries in the tensor, but the camera geometry that it encodes has only 18 degrees of freedom. This means that the trifocal tensor must satisfy 8 constraints, apart from scale ambiguity to make up the 27 degrees of freedom of a general $3 \times 3 \times 3$ tensor. The exact form of these constraints is not known precisely. Nevertheless, they must be enforced in order that the trifocal tensor should be well behaved. It will now be shown how this can be done, while minimizing algebraic error.

Formula for the Trifocal Tensor. We denote the three camera matrices P' and P'' by a_j^i and b_j^i respectively, instead of by $p_j'^i$ and $p_j''^i$. Thus, the three camera matrices P, P' and P'' may be written in the form $P = [I \mid 0]$, $P' = [a_j^i]$ and $P'' = [b_j^i]$.

In this notation, the formula for the entries of the trifocal tensor is :

$$T_i^{jk} = a_i^j b_4^k - a_4^j b_i^k \ . \tag{11}$$

Our task will be to compute a trifocal tensor T_i^{jk} of this form from a set of image correspondences. The tensor computed will minimize the algebraic error associated with the input data. The algorithm is quite similar to the one given for computation of the fundamental matrix. Just as with the fundamental matrix, the first step is the computation of the epipoles.

4.1 Retrieving the epipoles

We consider the task of retrieving the epipoles from the trifocal tensor. If the first camera has matrix $P = [I \mid \mathbf{0}]$, then the epipoles $\mathbf{e}_{21}$ and $\mathbf{e}_{31}$ are the last columns a_4^i and b_4^i of the two camera matrices $P' = [a_j^i]$ and $P'' = [b_j^i]$ respectively. These two epipoles may easily be computed from the tensor T_i^{jk} according to the following proposition.

Proposition 3. *For each $i = 1, \ldots, 3$, the matrix $T_i^{\cdot\cdot}$ is singular. Furthermore, the generators of the three left null-spaces have a common perpendicular, the epipole $\mathbf{e}_{21}$. Similarly epipole $\mathbf{e}_{31}$ is the common perpendicular of the right nullspaces of the three matrices $T_i^{\cdot\cdot}$.*

This proposition translates easily into an algorithm for computing the epipoles ([5, 3]). This algorithm may be applied to the tensor T_i^{jk} obtained from the linear algorithm to obtain a reasonable approximation for the epipoles.

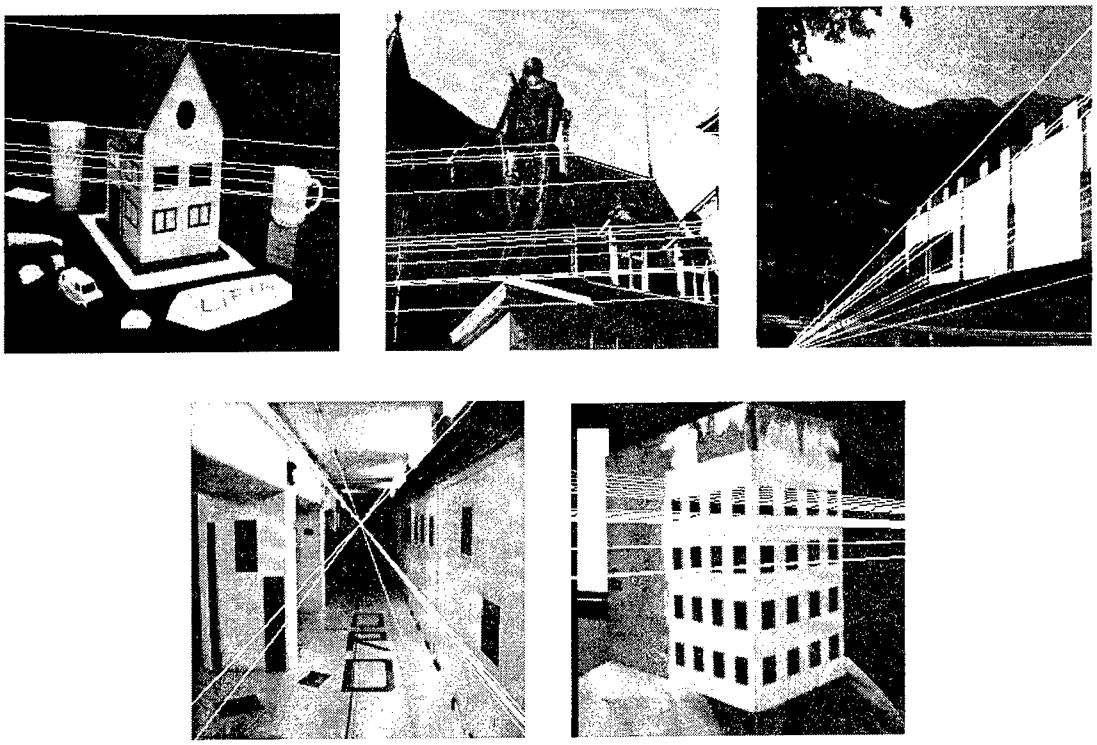


Figure 1: The images used in the experiments

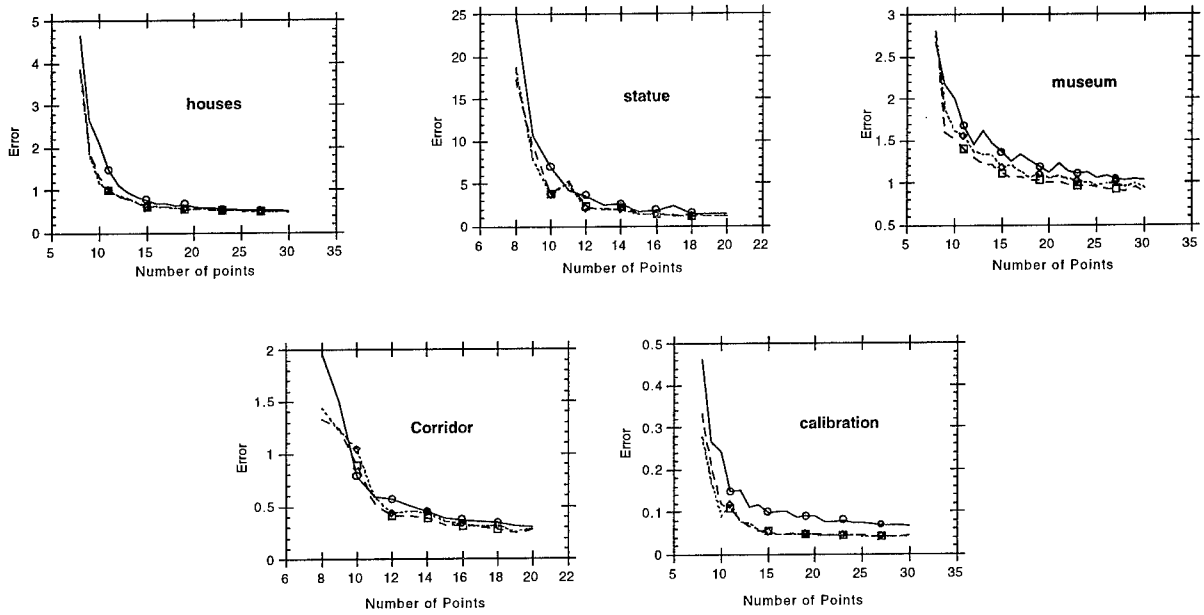


Figure 2: Results of the experimental evaluation of the algorithms. In each case, three methods of computing F were compared. In each graph, the top (solid) line shows the results of the normalized 8-point algorithm. Also shown are the results of minimizing geometric error and algebraic error, using the algorithm of this paper. In most cases, the result of minimizing algebraic error is almost indistinguishable from minimizing geometric error. Both are noticeably better than the non-iterative 8-point algorithm, though that algorithm gives reasonable results.

4.2 Constrained Estimation of the Trifocal Tensor

From the form (11) of the trifocal tensor, it may be seen that once the epipoles $\mathbf{e}_{21} = a_4^j$ and $\mathbf{e}_{31} = b_4^k$ are known, the trifocal tensor may be expressed linearly in terms of the remaining entries of the matrices a_i^j and b_i^k.

Assuming the epipoles a_4^j and b_4^k to be known, we may write $\mathbf{t} = H\mathbf{a}$ where $\mathbf{a}$ is the vector of the remaining entries a_j^i and b_j^i, $\mathbf{t}$ is the vector of entries of the trifocal tensor, and H is the linear relationship expressed by (11). We wish to minimize the algebraic error $||M\mathbf{t}|| = ||MH\mathbf{a}||$ over all choices of $\mathbf{a}$ constrained such that $||\mathbf{a}|| = 1$. The solution is the eigenvector corresponding to the least eigenvalue of $H^\top M^\top MH$.

In solving this set of equations to find $\mathbf{a}$ it is advisable to restrict the dimensionality of the solutions set by applying the constraint that $\sum_i a_4^i a_j^i = 0$ for each $j = 1, \ldots, 3$. This constraint is discussed in [5, 3]. Given that a_4^i is known, it is a linear constraint that may be expressed by a matrix equation $C\mathbf{a} = 0$. Thus, the minimization problem is to minimize $||MH\mathbf{a}||$ subject to the $||\mathbf{a}|| = 1$ and the linear constraint $C\mathbf{a} = 0$. This may be done by the algorithm given in [5, 3].

Writing $\hat{\mathbf{t}} = H\mathbf{a}$ where $\mathbf{a}$ is the solution vector, we see that $\hat{\mathbf{t}}$ minimizes algebraic error $||M\hat{\mathbf{t}}||$ subject to the condition that T_i^{jk} is of the correct form (11), for the given choice of epipoles.

Iterative Solution The two epipoles used to compute a correct constrained tensor T_i^{jk} are computed using the estimate of T_i^{jk} obtained from the linear algorithm. Analogous to the case of the fundamental matrix, the mapping $(\mathbf{e}_{21}, \mathbf{e}_{31}) \mapsto MH\mathbf{a}$ is a mapping $R^6 \to R^{27}$. An application of the Levenberg-Marquardt algorithm to optimize the choice of the epipoles will result in an optimal (in terms of algebraic error) estimate of the trifocal tensor. Note that the iteration problem is of modest size, since only 6 parameters, the homogeneous coordinates of the epipoles, are involved in the iteration problem.

This contrasts with an iterative estimation of the optimal trifocal tensor in terms of geometric error. This latter problem would require estimating the three camera paramters, plus the coordinates of all the points, a large estimation problem.

5 Conclusion

Experimental evidence backs up the assertion that minimizing algebraic distance can usually give good results at a fraction of the computation cost associated with minimizing geometric distance. The great advantage of the method for minimizing algebraic error given in this paper is that even for problems that need an iterative solution the size of the iteration problem is very small. Consequently, the iteration is very rapid and there is reduced risk of falling into a local minimum, or otherwise failing to converge.

The method has been illustrated by applying it to three problems. For the computation of the fundamental matrix, iteration is over only three homogeneous parameters. For the trifocal tensor, iteration is over 6 parameters. This leads to more efficient methods than have been previously known.

The general technique is applicable to problems other than those treated here. It may be applied in a straightforward manner to estimation of projective transformations between 2 or 3-dimensional point sets. In these problems, iteration is necessary if one restricts the class of available transformations to a subgroup of the projective group, such as planar homologies (used in [6]), or conjugates of rotations ([4]).

References

[1] Gene H. Golub and Charles F. Van Loan. *Matrix Computations, Second edition.* The Johns Hopkins University Press, Baltimore, London, 1989.

[2] R. I. Hartley. In defence of the 8-point algorithm. In *Proc. International Conference on Computer Vision*, pages 1064 – 1070, 1995.

[3] R. I. Hartley. A linear method for reconstruction from lies and points. In *Proc. International Conference on Computer Vision*, pages 882 – 887, 1995.

[4] Richard I. Hartley. Self-calibration from multiple views with a rotating camera. In *Computer Vision - ECCV '94, Volume I, LNCS-Series Vol. 800, Springer-Verlag*, pages 471–478, May 1994.

[5] Richard I. Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, to appear.

[6] A. Zisserman, D. A. Forsyth, J. L. Mundy, C. A. Rothwell, J. Liu, and N. Pillow. 3d object recognition using invariance. *AI Journal*, 78:239 – 288, 1995.

Robust Multi-Sensor Image Alignment*

Michal Irani
Dept. of Applied Math and CS
The Weizmann Institute of Science
76100 Rehovot, Israel

P. Anandan
David Sarnoff Research Center
Princeton, NJ 08543-5300, USA

Abstract

This paper presents a method for alignment of images acquired by sensor of different modalities. The paper has two main contributions: (i) It identifies an appropriate image representation for multi-sensor alignment, i.e., a representation which emphasizes the common information between the two multi-sensor images, suppresses the non-common information, and is adequate for coarse-to-fine processing. (ii) It presents a new alignment technique, which applies global estimation to *any choice of a local similarity measure*. In particular, it is shown that when this registration technique is applied to the chosen image representation with a local-normalized-correlation similarity measure, it provides a new multi-sensor alignment algorithm which is robust to outliers, and applies to a wide variety of *globally* complex brightness transformations between the two images.

Our proposed image representation does *not* rely on sparse image features (e.g., edge, contour, or point features). It is *continuous* and does not eliminate the detailed variations within local image regions. Our method naturally extends to coarse-to-fine processing, and applies even in situations when the multi-sensor signals are *globally* characterized by low statistical correlation.

1 Introduction

In images acquired by sensors of *different modalities*, the relationship between the brightness values of corresponding pixels is usually complex and unknown: Visual features present in one sensor image may not appear in the other image, and vice versa; contrast reversal may occur between the two images in some image regions, while not in others; multiple brightness values in one image may map to a single brightness value in the other image, and vice versa. In other words, the two images are usually not correlated in their entirety, i.e., they are not *globally* correlated (often, not even statistically correlated).

There are two fundamental questions that a multi-sensor alignment algorithm should address: (i) What is a good image representation to work with (i.e., what representation will bring out the common information between the two multi-sensor images, while suppressing the non-common information)? (ii) What is an appropriate similarity measure for matching the two images within the selected representation?

Previous work on multi-sensor image alignment (e.g., [Dana and Anandan, 1993, Kumar *et al.*, 1994, van den Elsen and Viergever, 1994, Li *et al.*, 1995, Li and Zhou, 1995, Viola and Wells III, 1995]) can broadly be classified into two major classes of algorithms. These classes differ in the way they address the two abovementioned questions:

1. Methods that use an *invariant image representation*. By invariant image representation we refer to a representation that is invariant to changes in brightness and contrast, as well as to contrast reversal. Some examples of invariant image representations are edge maps [Dana and Anandan, 1993], oriented edge vector fields [Kumar *et al.*, 1994], contour features [Li *et al.*, 1995], and feature points [Li and Zhou, 1995]. Such representations aim at increasing the visual similarity between of the two images. Once this is achieved, registration techniques that assume similar appearance (e.g., that are based on the *brightness constancy assumption*) can be applied. For example, the registration methods employed in [Dana and Anandan, 1993,

*This work was supported by NASA-Ames Research Center under contract NAS2-14301

Kumar *et al.*, 1994] are extensions of the direct gradient-based registration methods [Bergen *et al.*, 1992, Irani *et al.*, 1994]).

However, in the process of creating an invariant image representation, important image information is usually lost. For example, in [Dana and Anandan, 1993, Kumar *et al.*, 1994, Li *et al.*, 1995] there is a thresholding step. This step usually eliminates most of the detailed variations within local regions of the images, leaving only a *sparse* set of highly significant image features. Moreover, the choice of threshold is very data and sensor dependent.

2. Methods that use an *invariant similarity measure* to register the multi-sensor images, and therefore do not require an invariant image representation.

An example of such a similarity measure is *Mutual Information* [Viola and Wells III, 1995], which is a measure of the statistical correlation between two images. The method suggested by [Viola and Wells III, 1995] is applied directly to the raw multi-sensor intensity images, and does not require an invariant image representation. This method assumes, however, that the statistical correlation between the two images is *global*, an assumption which is often violated (e.g., Figure 4). Moreover, the statistical correlation between raw multi-sensor images tends to decrease with the reduction in spatial resolution (Section 2). Therefore, [Viola and Wells III, 1995] in its current form does not naturally extend to coarse-to-fine estimation, which is often used to handle large misalignments. These issues will be referred to in Section 2.

In order to address the issues mentioned above, we have developed an approach which uses a *locally* invariant similarity measure while globally *constraining* the local matches. In particular, our approach to multi-sensor image alignment does *not* assume global correlation (regular or statistical) of the images, but only a local one. The underlying chosen image representation is *continuous*, and avoids thresholding and hence loss of image detail. The representation is invariant to contrast reversal, provides orientational sensitivity, and is suitable for coarse-to-fine processing. The estimation process has a built-in outlier rejection mechanism, which is critical to multi-sensor alignment due to the plurality of non-common image features across the two images (as a matter of fact, in many situations there are more "outliers" than "inliers" in a multi-sensor image pair). The motion models used in this work were 2D parametric transformations. The algorithm, however, can be extended to 3D motion models as well.

The rest of the paper is organized as follows: Section 2 describes the chosen image representation. Section 3 describes the global alignment method with a local similarity measure. Section 4 presents results of applying our algorithm to IR/EO image pairs.

2 The Image Representation

The underlying assumption of multi-resolution alignment is that the corresponding signals at all resolution levels contain enough correlated structure to allow stable matching. This assumption is generally true when an image pair is obtained by the *same sensor*, or by two different cameras of *same modality*. However, in multi-sensor image pairs (i.e., image pairs taken by sensors of *different* modalities), the signals are correlated primarily in high resolution levels, while correlation between the signals tends to degrade substantially with the reduction in spatial resolution. This is because high resolution images capture high spatial frequency information, which corresponds to physical structure of the scene that is common to the two images. Low resolution images, on the other hand, depend heavily on illumination and on the photometric and physical imaging properties of the sensors (which are characterized by low frequency information), and these are substantially different in two multi-modality images.

To capture the common scene detail information while suppressing the non-common illumination and sensor-dependent properties, the images are transformed into high-pass *energy* images (e.g., see [Burt, 1988]). An example of such an energy image is a *Laplacian-energy image*, which is formed by first high-pass filtering the image with a Laplacian filter, then squaring it. This facilitates coarse-to-fine search based on *signal details*. In [Burt, 1988] the Laplacian-energy image is used for effectively detecting small (high-resolution) temporal changes already at low resolution levels.

High-pass energy image representations are useful for multi-sensor alignment, because:
(i) The creation of such energy images does not involve any thresholding, and therefore preserves all image detail. This is in contrast to "invariant" representations (e.g., edge maps [Dana and Anandan, 1993], edge vectors [Kumar *et al.*, 1994], contours [Li *et al.*, 1995], point features [Li and Zhou, 1995]), which eliminate most of the detailed variations within local image regions.
(ii) The image information which is eliminated in the creation of the high-pass energy images is exactly that which is *not* common to the two multi-sensor images. In particular: (a) the sensor-dependent

low-resolution information is eliminated, and (b) contrast-reversal which may occur between the sensors (e.g., Fig. 3) is removed by the squaring operation. In other words, the energy image representation is *invariant to contrast reversal.*
(iii) As mentioned in [Burt, 1988], a pyramid data structure of the high-pass energy image projects high resolution signal information into low resolution levels. In our case, this facilitates coarse-to-fine alignment based on *correlated* scene details, as opposed to using pyramids of the raw multi-sensor images (which contain *uncorrelated* sensor information at low spatial resolutions).

However, the Laplacian, being a rotationally invariant operator, does not preserve directional information. This leads to potential false correspondences of patterns that are oriented along different directions in the Laplacian energy images. The energy-image representation that we use is based on *directional-derivative* filters rather than a Laplacian filter. On top of the abovementioned advantages of high-pass energy images, the *directional-derivative*-energy also preserve *directional* information, and thereby avoid this problem. This further enhances the robustness of the registration algorithm against the numerous outliers so common in a multi-sensor image pair.

The directional derivative filter is applied to the raw image in four directions (horizontal, vertical, and the two diagonals). Then, each of the four generated derivative images is squared. (Since the squaring operation doubles the frequency band, the raw image is filtered with a Gaussian prior to the derivative filtering, to avoid aliasing effects).

The alignment algorithm (Section 3) is applied *simultaneously* to all 4 corresponding multi-sensor pairs of directional-derivative-energy images, seeking a *single* parametric transformation $\vec{p}$, which *simultaneously* brings all pairs into alignment (see Section 3).

Fig. 1 shows an example of the four directional-derivative-energy pairs constructed from a multi-sensor image pair. Fig. 2 shows the Gaussian pyramid constructed for one of the four multi-sensor *pairs* of directional-derivative-energy images.

3 The Alignment Algorithm

To align the multi-band energy image representation (Section 2), our alignment algorithm uses a *local* correlation-based similarity measure, *without* assuming global correlation (regular or statistical) between the images. We have applied the algorithm with a normalized-correlation-based local similarity measure for reasons explained below. However, it can be similarly applied with a *local* statistical-correlation-based similarity measure (e.g., based on Mutual Information), or *any other appropriate local measure.*

The global parametric estimation is applied *directly* to the collection of all local *correlation surfaces*, while avoiding an independent local search for peaks in the individual surfaces. Global alignment has the advantage of directly estimating the global parametric transformation, without first committing to any particular matches locally. In other words, local matching is constrained by global alignment. Such a scheme is useful in any alignment algorithm, but is particularly critical in multi-sensor alignment, due to the plurality of outliers across sensors and hence the unreliability of local matches. Although global alignment has been previously used for image alignment (e.g., [Bergen *et al.*, 1992, Irani *et al.*, 1994]), they have relied on "brightness constancy", which is severely violated in a multi-sensor image pair (as well as in their energy images). In this work, we have generalized global alignment to use *any* local similarity measure (e.g., normalized correlation) which is suitable for multi-sensor energy-image alignment. This is done via global regression applied *directly* to the local *similarity-measure surfaces* (e.g., correlation surfaces), as described in Section 3.1.

Global alignment is particularly critical when using the directional-derivative images: no prior local estimation process can produce meaningful local matches on a directional-derivative image pair, as these images lack information in the direction *perpendicular* to the directional derivative (the "aperture problem"). The *simultaneous* and *global* registration of all (four) directional pairs, however, provides full directional information.

Motion Models: When the scene can be approximated by a planar surface, or when the baseline between the two sensors is small relative to their distance from the scene, then the displacement field between the two images can be modeled in terms of a single 2D parametric transformation (see [Bergen *et al.*, 1992] for a taxonomy of motion models).

We have focused our attention on alignment using a 2D parametric transformation, although our approach generalizes to 3D models as well. Specifically, we focus on parametric transformations which are *linear* in their unknown parameters $\{p_i\}$. For such transformations, the motion vector $\vec{u}(x,y) = (u(x,y), v(x,y))^T$ can be expressed as:

$$\vec{u}(x,y;\vec{p}) = X(x,y) \cdot \vec{p}, \qquad (1)$$

where $X(x,y)$ is a matrix which depends only on the pixel coordinates (x,y), and $\vec{p} = (p_1, ..., p_n)^T$ is

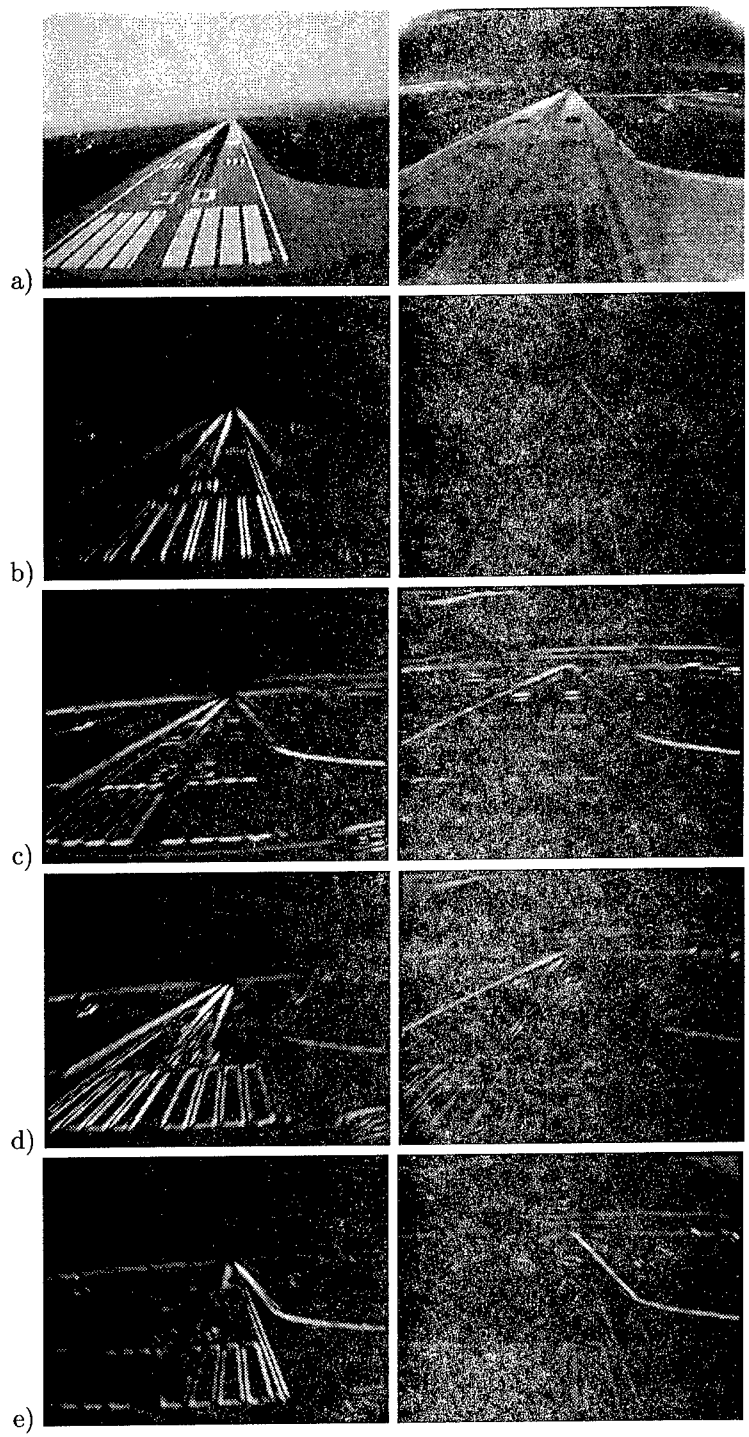


Figure 1: The four directional-derivative-energy image pairs. *Left column:* EO. *Right column:* IR. (a) The raw multi-sensor image pair. (b) horizontal derivative energy, (c) vertical derivative energy, (d,e) energies of diagonal derivatives.

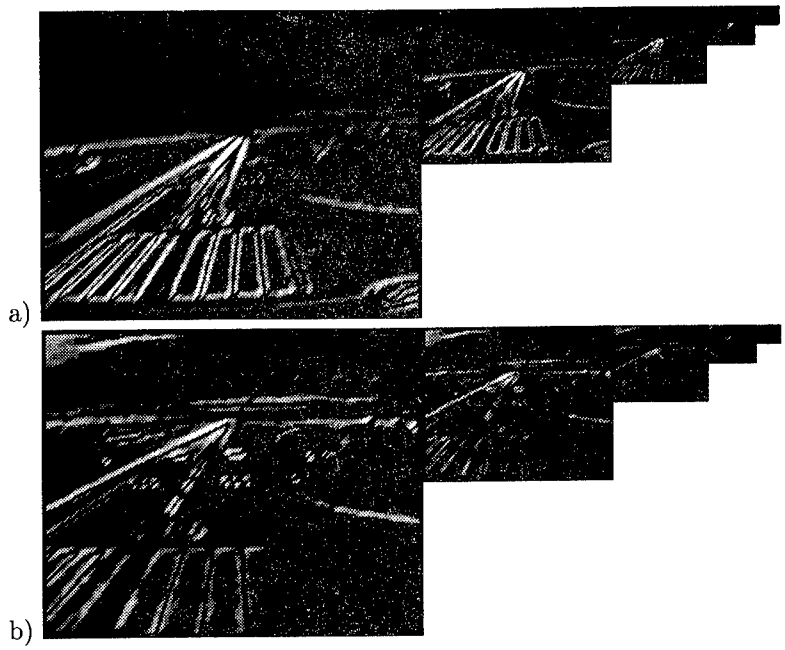


Figure 2: The Gaussian pyramid constructed for one of the four pairs of directional-derivative-energy images (Fig. 1.d): (a) EO. (b) IR.

the parameter vector. For example, for an *affine* transformation:

$$\begin{bmatrix} u(x,y;\vec{p}) \\ v(x,y;\vec{p}) \end{bmatrix} = \begin{bmatrix} p_1 + p_2 x + p_3 y \\ p_4 + p_5 x + p_6 y \end{bmatrix}, \qquad (2)$$

therefore, in this case: $\vec{p} = (p_1, p_2, p_3, p_4, p_5, p_6)^T$ and

$$X = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix},$$

and for a *quadratic* transformation:

$$\begin{bmatrix} u(x,y;\vec{p}) \\ v(x,y;\vec{p}) \end{bmatrix} = \begin{bmatrix} p_1 + p_2 x + p_3 y + p_7 x^2 + p_8 xy \\ p_4 + p_5 x + p_6 y + p_7 xy + p_8 x^2 \end{bmatrix},$$

therefore: $\vec{p} = (p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8)^T$ and

$$X = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 & x^2 & xy \\ 0 & 0 & 0 & 1 & x & y & xy & y^2 \end{bmatrix}.$$

The Normalized-Correlation as a Local Similarity Measure: Normalized-correlation of two signals is invariant to local changes in mean and contrast. In other words, when the two signals are *linearly related*, their normalized-correlation is 1. When the linear relationship does not hold, but the two signals contain *similar spatial variations* (as measured in the form of local fluctuations), the normalized-correlation will still give a value close to unity.

In general, however, the global relationship between two multi-sensor images is complex, and therefore the two signals are not globally correlated (even after computing the energy images). Statistical correlation is a better *global* measure than regular or normalized correlation, but may still not be a strong enough global similarity measure, because multiple brightness values in one image may map to a single brightness value in the other image, and vice versa. *Locally*, however, within *small* image patches which contain corresponding image features, statistical correlation is high. Normalized-correlation is a linear approximation of the statistical correlation of two signals in a small window, and is cheaper to compute.

The *energy* images that we compute tend to highlight the local variations that correspond to local structure in the scene. These images are invariant to contrast reversal, but vary in mean and contrast. When the relationship between corresponding patches deviates from linear, the normalized-correlation (applied over local windows) is less than 1, but is still high for the correct displacement. For other displacements the normalized-correlation will be low, especially for highly textured image patches. Therefore, the local normalized-correlation surface of such patches will be concave with a prominent peak at the correct displacement. For corresponding image patches that contain *mutually exclusive* image features (i.e., image features which appear in only one of the 2 multi-sensor images – a thing which occurs frequently), the local correlation surface will not have a concave shape with a prominent peak. Therefore, the *structure* of the local-

normalized-correlation *surfaces* provides useful information for alignment. The information from all of these local structures, however, should be *simultaneously* used to determine the global alignment parameters. This is essential to avoid the numerous potential false matches in limited local analysis. This is achieved via global regression applied *directly* to the collection of local normalized-correlation surfaces, as described in Section 3.1.

3.1 Global Alignment with Local Correlation

Given two images, f and g, and their directional-derivative energy images, $\{f_i\}_{i=1}^4$ and $\{g_i\}_{i=1}^4$, find the parametric transformation $\vec{p}$ which maximizes the sum of all local normalized-correlation values. Let $S_i^{(x,y)}(u,v)$ denote a correlation *surface* corresponding to a pixel (x,y) in f_i. For any shift (u,v) of g_i relative to f_i, $S_i^{(x,y)}$ is defined as:

$$S_i^{(x,y)}(u,v) \stackrel{\text{def}}{=} f_i(x,y) \circ_{\mathbf{N}} g_i(x+u, y+v)$$

where $\circ_{\mathbf{N}}$ denotes normalized correlation computed over a small window. Let $\vec{u} = (u(x,y;\vec{p}), v(x,y;\vec{p}))$ denote the motion field described by the parametric transformation $\vec{p}$. Then the parametric registration problem can be stated as follows: Find the parametric transformation $\vec{p}$ that maximizes the *global* similarity-measure $M(\vec{p})$:

$$\begin{aligned} M(\vec{p}) &= \Sigma_{x,y}\Sigma_i S_i^{(x,y)}(u(x,y;\vec{p}), v(x,y;\vec{p})) \\ &= \Sigma_{x,y}\Sigma_i S_i^{(x,y)}(\vec{u}(x,y;\vec{p})). \end{aligned} \quad (3)$$

To solve for $\vec{p}$ that maximizes $M(\vec{p})$, we use Newton's method [Luenberger, 1984], which iteratively fits quadratic approximations to the objective function, and refines the peak location that maximizes these quadratic surfaces. In order to provide the context for our use of Newton's method for the maximization problem at hand, we first briefly outline the steps of this method.

Given the current estimate of the motion parameters $\vec{p_0}$, let

$$M(\vec{p}) = M(\vec{p_0}) + (\nabla_{\vec{p}} M(\vec{p_0}))^T \vec{\delta_p} + \vec{\delta_p}^T H_M(\vec{p_0}) \vec{\delta_p} \quad (4)$$

denote the quadratic approximation of $M(\vec{p})$ around $\vec{p_0}$, where, $\vec{\delta_p} = \vec{p} - \vec{p_0}$ is the unknown refinement step of $\vec{p_0}$ that we want to solve for, $\nabla_{\vec{p}} M$ denotes the gradient of M, and H_M denotes the Hessian of M (i.e., the matrix of second derivatives), both computed around $\vec{p_0}$. According to Newton's method [Luenberger, 1984], the refinement $\vec{\delta_p}$ computed based on this approximation is:

$$\vec{\delta_p}^* = -(H_M(\vec{p_0}))^{-1} \cdot \nabla_{\vec{p}} M(\vec{p_0}) \quad (5)$$

To apply the Newton's refinement step to our problem, we derived the expressions for $\nabla_{\vec{p}} M$ and H_M *in terms of the measurable image quantities*, i.e., the collection of correlation surfaces $\{S_i^{(x,y)}\}$: Using the chain-rule of differentiation, we obtain

$$\begin{aligned} \nabla_{\vec{p}} M(\vec{p}) &= \Sigma_{x,y,i} \nabla_{\vec{p}} S_i(\vec{u}) = \Sigma_{x,y,i} (X^T \cdot \nabla_{\vec{u}} S_i(\vec{u})) \\ H_M(\vec{p}) &= \Sigma_{x,y,i} (X^T \cdot H_{S_i}(\vec{u}) \cdot X) \end{aligned} \quad (6)$$

where X is the matrix defined in Eq. (1), $\nabla_{\vec{u}} S_i$ is the gradient of $S_i^{(x,y)}(\vec{u})$, and H_{S_i} is the Hessian of $S_i^{(x,y)}(\vec{u})$.

In other words, the quadratic approximation of M around $\vec{p_0}$ is obtained by combining the quadratic approximations of each of the local correlation surfaces $\{S_i^{(x,y)}\}_{x,y,i}$ around the *local* displacement vector $\vec{u_0} = \vec{u}(x,y;\vec{p_0})$, which is induced at pixel (x,y) by the parametric transformation $\vec{p_0}$ (estimated at the previous iteration).

Substituting Eqs. (6) into Eq. (5) provides an expression for the refinement step $\vec{\delta_p}^*$ in terms of the correlation surfaces $\{S_i^{(x,y)}\}$:

$$\vec{\delta_p}^* = -(\Sigma_{x,y,i}\, X^T H_{S_i}(\vec{u_0}) X)^{-1} \, (\Sigma_{x,y,i}\, X^T \nabla_{\vec{u}} S_i(\vec{u_0})) \quad (7)$$

Note that these steps do not make any assumptions about the local correlation surface, except that it is twice differentiable. Thus, *any local similarity-measure* can be substituted for correlation, and our method will still apply.

The steps of the algorithm: To account for large misalignments between pairs of images, we perform multi-resolution coarse-to-fine estimation, e.g., as in [Bergen *et al.*, 1992]. A Laplacian (or a Gaussian) pyramid is constructed for each of the energy images. Let f_{il} and g_{il} $(i = 1,2,3,4)$ denote the directional-derivative energy images at resolution level l in the pyramids of f_i and g_i, respectively. Starting at the coarsest resolution level with $\vec{p_0}$ initially set to 0, the following steps are performed at each resolution level:

1. For each pixel (x,y) at f_{il} $(i = 1,2,3,4)$, compute a local normalized-correlation surface around the displacement $\vec{u_0}$ (i.e., around the displacement estimated at the previous iteration). In practice, the correlation surface is estimated only for a small number of displacements $\vec{u}$ of g_{il} *within a radius* d around $\vec{u_0}$, i.e.:

$$S_{l_i}^{(x,y)}(\vec{u}) = f_{il}(x,y) \circ_{\mathbf{N}} g_{il}(x+u, y+v) \ ,$$

$$\forall \vec{u} = (u,v) \ s.t. \ ||\vec{u} - \vec{u_0}|| \leq d$$

where the radius d is determined by the size of the masks used for discretely estimating the first and second order derivatives of $S_i^{(x,y)}(\vec{u})$ at $\vec{u_0}$. In our current implementation we used Beaudet's masks [Beaudet, 1978] to estimate the first and second order derivatives of the surfaces. We have experimented both with 3×3 masks (i.e., $d = 1$) and with 5×5 masks (i.e., $d = 2$).

2. Perform the regression step of Eq. (7) to compute the parametric refinement $\vec{\delta_p}^*$.

3. Update $\vec{p_0}$: $\vec{p_0} := \vec{p_0} + \vec{\delta_p}^*$, and go back to step 1.

After repeating the above process for a few iterations (typically 4), the parameters $\vec{p}$ are propagated to the next resolution level, and the process is repeated at that resolution level. The process is stopped when the iterative process at the highest resolution level is completed.

In practice, to improve performance, we add an *image warping* step before each iteration (as in [Bergen *et al.*, 1992]). The inspection images $\{g_i\}$ are warped towards the reference images $\{f_i\}$ according to the current estimated parametric transformation $\vec{p_0}$. After warping the images, $\vec{p_0}$ is set to 0, and $\vec{\delta_p}^*$ is estimated between the pairs of references and warped inspection images. Warping compensates for the spatial distortions between the pairs of images (e.g., scale difference, rotations, etc), and hence improves the quality of the correlation.

Outlier rejection: To further condition and robustify the regression step of Eq. (7), only pixels (x, y) for which the quadratic approximation of $S_i^{(x,y)}(\vec{u})$ around $\vec{u_0}$ is *concave* are used in the regression process. Other pixels are ignored. Since corresponding multi-sensor image patches which have *mutually exclusive* image features will not tend to have a concaved-shaped local correlation surfaces, they will be eliminated from the regression at this point. Moreover, the contribution of each pixel to the regression step is weighted by the determinant of it Hessian. This built-in *outlier rejection* mechanism provides the algorithm with a strong *locking* property onto a dominant parametric motion, even in the presence of independent motions, noise, and exclusive features that appear in only one of the sensor-images (but not in the other).

4 Experimental Results

The alignment algorithm described in Section 3 was implemented and applied with an affine parametric model (Eq. 2) to pairs of multi-sensor images. Fig. 3 shows result of alignment of two multi-sensor images (visible and IR) obtained by sensors mounted on an aircraft approaching landing. Note the significant difference in scale between the two images (due to significantly different internal sensor parameters). Also note that contrast reversal occurs in some parts of the images (e.g., runway markings), while not in others (e.g., runway boundaries).

The algorithms has been applied successfully even in very challenging situations, such as the one shown in Fig 4. Note the significant difference in image content between the two sensor-images. Apart from having significantly different appearance, there are many non-common features (i.e., *outliers*) in the multi-sensor image pair, which can theoretically lead to false matches. These are overcome by the built-in outlier mechanism of our algorithm (see Section 3).

References

[Beaudet, 1978] Paul R. Beaudet. Rotationally invariant image operators. In *International Conference on Pattern Recognition*, pages 579–583, 1978.

[Bergen *et al.*, 1992] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, pages 237–252, Santa Margarita Ligure, May 1992.

[Burt, 1988] P.J. Burt. Smart sensing with a pyramid vision machine. *Proceedings of the IEEE*, 76:1006–1015, 1988.

[Dana and Anandan, 1993] K. J. Dana and P. Anandan. Registration of visible and infrared images. In *Proc. SPIE Conf. on Arch., Hardware and FLIR in Auto. Targ. Rec.*, pages 1–12, 1993.

[Irani *et al.*, 1994] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, January 1994.

[Kumar *et al.*, 1994] R. Kumar, K. Dana, and P. Anandan. Frameless registeration of mr and ct 3d volumetric data sets. In *Proc. of the Workshop on Applications of Computer Vision II*, Sarasota, Fl., 1994.

[Li and Zhou, 1995] H. Li and Y.T. Zhou. Automatic eo/ir sensor image registration. In *IEEE Int. Conf. on Image Proc.*, volume B, pages 161–164, 1995.

[Li *et al.*, 1995] H. Li, B.S. Manjunath, and S.K. Mitra. A contour-based approach to multisensor image registration. *IEEE Trans. on Image Processing*, pages 320–334, 1995.

[Luenberger, 1984] David G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, Reading, MA., 1984.

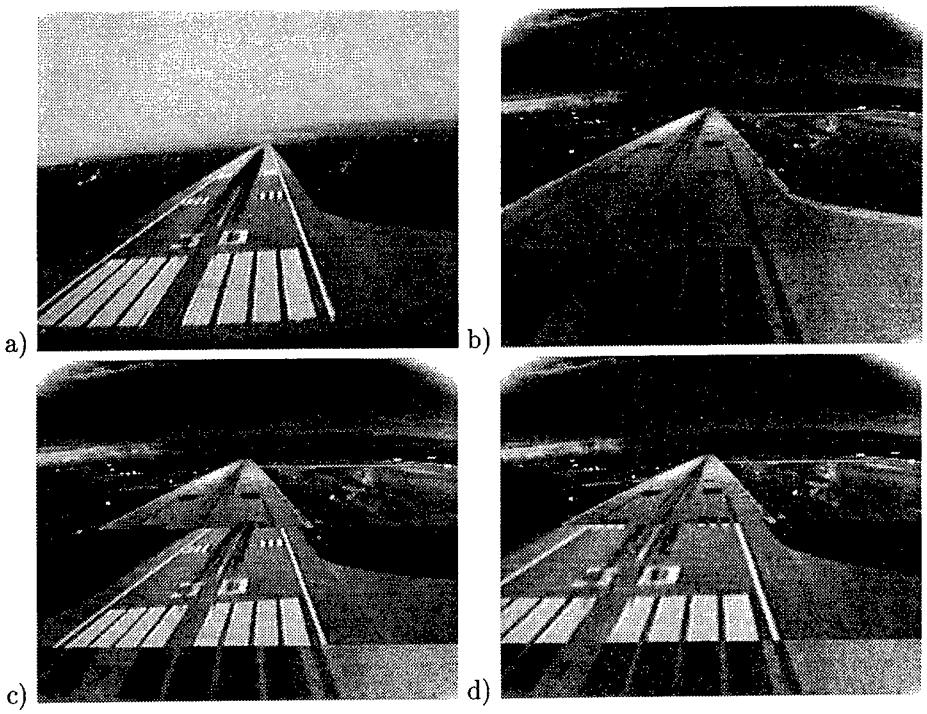


Figure 3: Multi-sensor Alignment.
(a) EO image. (b) IR image. (c) Composite display of the two multi-sensor images *before* alignment. Horizontal strips from the two images are spliced together. Note the significant misalignments between the images (e.g., the runway markings and the borders of the runway). (d) Composite (spliced) display of the two multi-sensor images *after* alignment. Note that all structures in the scene are aligned.

[van den Elsen and Viergever, 1994] P. A. van den Elsen and M.A. Viergever. Marker-guided multi-modality matching of the brain. *European Radiology*, 4(1):45–51, 1994.

[Viola and Wells III, 1995] P. Viola and W. Wells III. Alignment by maximization of mutual information. In *International Conference on Computer Vision*, pages 16–23, Cambridge, MA, June 1995.

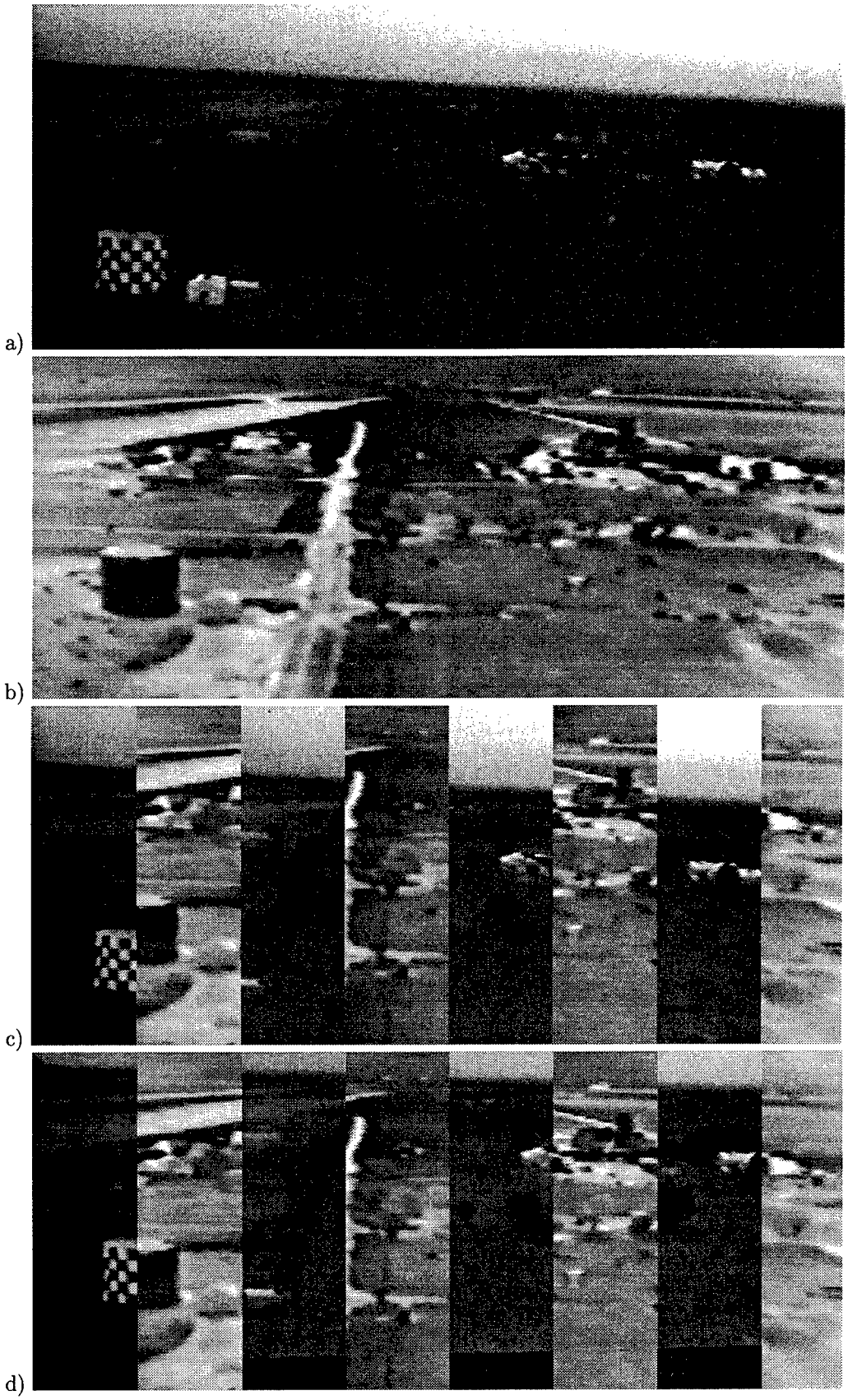


Figure 4: Multi-sensor Alignment.
(a) EO image. (b) IR image. (c) Composite (spliced) display *before* alignment. (d) Composite (spliced) display *after* alignment. Note in particular the perfect alignment of the water-tank at the bottom left of the images, the building with the arched-doorway at the right, and the roads at the top left of the images.

The Cubic Rational Polynomial Camera Model

Richard I. Hartley
G.E. Corporate R & D
1 Research Circle
Niskayuna, NY 12309

Tushar Saxena
CMA Consulting
1400 Balltown Rd
Niskayuna, NY 12301

Abstract

This paper describes an implementation of the Cubic Rational Polynomial Camera model developed as part of the FOCUS project. FOCUS ([1]) is on ongoing "shared vision" IR&D project jointly sponsored by Lockheed Martin Missiles and Space (LMMSS/Sunnyvale) and General Electric CR&D. A cubic camera has the advantage that all cameras, such as projective, affine and the linear pushbroom, which map the image points as rational polynomial functions (of degree no greater than 3) of the coordinates of a world point, can be treated as special cases of the cubic camera. This paper demonstrates that the cubic camera can very effectively model even those cameras which express the image points as complicated functions of world coordinates, such as radicals. In particular, it is empirically demonstrated that a SAR sensor is very accurately approximated by a cubic camera, but not by any linear camera model.

The paper also outlines an algorithm for estimating the parameters of the cubic camera, given a set of image to world correspondences. The non-linear nature of this camera can make parameter estimation a very unstable process. The slightest noise in the coefficients of the nonlinear terms can lead to a completely unrealistic model of the camera. This paper discusses some refinements such as avoiding degeneracies, data normalization, and regularization which are necessary for accurate estimation of the cubic camera parameters and minimization of noise in the coefficients of the higher degree terms.

1 Introduction

A basic requirement of the FOCUS ([1]) project is to be able to compute camera models and do model building using complex and general camera models. To this purpose, a Cubic Rational Polynomial camera model has been developed in FOCUS to aid in these tasks. Lockheed Martin and GE initiated the FOCUS project in January 1996 using GE's TargetJr as the MSE/IU platform. Funding for FOCUS at GE CR&D is provided by the Lockheed Martin Corporation. We acknowledge that this work would not have been possible without Government sponsored camera modelling and IU technology during the past decade.

A cubic camera models the coordinates of the image point as ratios of third degree polynomials in the coordinates of the world point. Given a set of image-world correspondences, the objective of the cubic camera estimation problem is to determine the set of coefficients (a total of 80) of the polynomials in the cubic camera model such that the error with which the camera maps the world points to the image points in this set of correspondences, is minimized. In this paper, we outline an algorithm which solves the cubic camera estimation problem by applying a least squares minimization to make an initial *guess* of the camera model, and then iteratively refining that guess and minimizing the error using a method based on Levenberg-Marquardt algorithm.

Due to the existence of non-linear terms in the camera model, even a small noise in the coefficients of the higher degree terms can lead to a large amount of error. A related problem is that of extrapolation. Since the solution of the camera model is not unique, there may exist models which produce a small error in the given set of correspondences, but assign such values to the coefficients of the higher degree terms which produce a completely unrealistic mapping of points outside the given set of correspondences. This leads to complications while extrapolating the model to points outside the given correspondence set. To overcome this such problems which are unique to non-linear camera models, we use the techniques of Data Normalization and Regularization. Specifically, we demonstrate that constraining the coefficients of the nonlinear terms to be as small as possible, generates a more realistic camera model which extrapolates better on the points outside the data set.

It is easy to see that all linear cameras such as the affine, perspective, and linear pushbroom cameras, can be viewed as special cases of the cubic camera. However, it is not so straightforward to use the cubic camera estimation for this special cases. Some of the problems encountered are the same as before, namely those concerning the unstability of higher degree terms leading to *over-parametrization*, which basically estimates a non-linear approximation to a completely linear camera. We

demonstrate how the techniques such as regularization can also be exploited to overcome this problem, and get a more linear approximation in these special cases.

Finally, we conjecture that rational polynomial cameras indeed provide a very accurate approximation of even the *non-polynomial* cameras. In particular, we consider the SAR sensor, which models the image point as complicated radical functions of the coordinates of the world point. We provide empirical evidence which demonstrates that the cubic camera provides an extremely accurate approximation of this sensor, despite the fact that SAR is not a rational polynomial camera. The cubic-approximation of SAR is compared to the perspective and linear pushbroom approximations of the same. Our empirical results show that the cubic-approximation performs at least four orders of magnitude better.

2 The Cubic Camera Model

The cubic Rational Polynomial (RP) camera provides an abstraction of many types of camera models. The essential aspect of a camera model is the manner in which it maps points in space to points in an image. In the case of the RP camera, this mapping can be expressed in terms of rational polynomial functions of the world-coordinates of the object point.

Thus, the mapping defined by an RP camera is of the form

$$u = N_u(\mathbf{x})/D_u(\mathbf{x}) v = N_v(\mathbf{x})/D_v(\mathbf{x}) \qquad (1)$$

where $\mathbf{x} = (x, y, z, t)^\top$ is the homogeneous coordinate of a 3D point, $(u, v)^\top$ is the corresponding image point, and N_u, D_u, N_v and D_v are homogeneous polynomials of degree n.

A general homogeneous polynomial of degree n in r variables contains $\binom{n+r-1}{n} = \frac{(n+r-1)!}{n!(r-1)!}$ terms. In the particular case of polynomials in the coordinates of $\mathbf{x}$ we have $r = 4$ and so the number of terms is $n(n+1)(n+2)/6$.

We will consider most particularly the case in which $n = 3$ and refer to this as the Cubic camera. Each of the polynomials $N_u(\mathbf{x})$, $D_u(\mathbf{x})$, $N_v(\mathbf{x})$ and $D_v(\mathbf{x})$ has 20 terms, and hence may be parametrized by 20 coefficients. This amounts to a total of 80 parameters in all. It may be noted that in some descriptions of the Cubic camera, each of the coordinates x, y and z as well as the image coordinates u and v is subject to a scaling and offset, which adds an extra 10 parameters. However, these extra transformations may be incorporated into the rational cubic polynomial mappings, and are hence non-essential. They will be ignored in this exposition.

The polynomials N_u, D_u, N_v and D_v are homogeneous polynomials in the coordinates x, y, z and t of the 3D points. This means that each of the terms has the same degree, in this case 3. This is done so that the mapping is not dependent on the particular representation of the point $\mathbf{x}$ as a homogeneous vector. It is possible to dehomogenize the polynomials by setting $t = 1$. In this case the terms of the polynomials will have different degrees, and we can talk of constant, linear, quadratic, cubic terms. Whenever we talk of the degree of a term of a polynomial, or of the corresponding coefficient, it is this dehomogenized degree that will be meant.

3 Special Cases of the Cubic Camera

Many of the common cameras may be considered as special cases of the Cubic camera.

Projective Camera. The projective camera is defined by a mapping $(wu, wv, w)^\top = P\mathbf{x}$ where P is a 3×4 matrix, u and v are the image coordinates, and w is an unknown scale. This may also be written as

$$u = \frac{\mathbf{p}_1{}^\top \mathbf{x}}{\mathbf{p}_3{}^\top \mathbf{x}}$$
$$v = \frac{\mathbf{p}_2{}^\top \mathbf{x}}{\mathbf{p}_3{}^\top \mathbf{x}} .$$

Thus, we see that this is a special case of the RP camera in which $N_u(\mathbf{x})$, $D_u(\mathbf{x})$, $N_v(\mathbf{x})$ and $D_v(\mathbf{x})$ are linear functions and $D_u = D_v$.

Linear Pushbroom Camera. The linear pushbroom camera described in [5] is an example of an RP camera. The linear pushbroom camera is an approximation of the camera model represented by a SPOT satellite pushbroom sensor. The defining equation is $(u, wv, w)^\top = P(x, y, z, 1)^\top$ where as before, P is a 3×4 matrix, u and v are the image coordinates, and w is an unknown scale. In terms of a homogeneous object point $\mathbf{x} = (x, y, z, t)^\top$, this may be written as

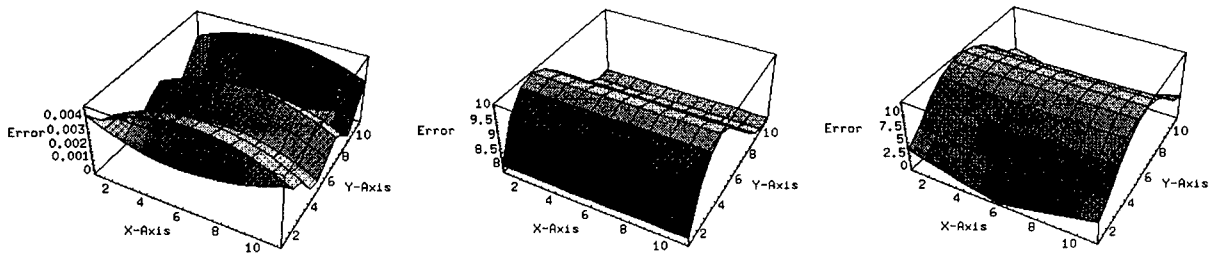
$$u = \frac{\mathbf{p}_1{}^\top \mathbf{x}}{t}$$
$$v = \frac{\mathbf{p}_2{}^\top \mathbf{x}}{\mathbf{p}_3{}^\top \mathbf{x}}$$

where $\mathbf{x} = (x, y, z, 1)^\top$. In this case, it is equivalent to an RP camera with $N_u(\mathbf{x})$, $N_v(\mathbf{x})$ and $D_v(\mathbf{x})$ linear functions, and $D_u(\mathbf{x}) = t$.

Affine Camera. The affine camera is a special case of the projective camera in which the camera matrix has a special form in which the last row is $(0, 0, 0, 1)$. This may be modelled as a Cubic camera for which

$$u = \frac{\mathbf{p}_1{}^\top \mathbf{x}}{t}$$
$$v = \frac{\mathbf{p}_2{}^\top \mathbf{x}}{t}$$

SAR images. SAR sensors may be approximated with the Cubic camera with excellent accuracy. This was demonstrated by testing the Cubic model against some synthetic correspondence data constructed as follows. Consider a SAR sensor moving in the x axial direction at an altitude of 3000m above a nominal ground plane and imaging a section of the ground at distances between 5000 and 7000 metres to the side of the flight path. Points were chosen over a 2000m $\times$ 2000m swath of ground at altitudes between -500m and 500m, and their corresponding image coordinates were computed, assuming a 1m pixel, thus creating a 2000 $\times$ 2000 pixel image. Thus, the u coordinate in the image of a point



Cubic model LP model Perspective model

Figure 1: *Fitting error for synthetic SAR data using Cubic, Perspective and Linear Pushbroom models. The graphs show the error for points in the plane $z = 0$, but for fitting, data points at altitdues between -500m and 500m were used. The average error for Perspective and LP sensors was approximately 6 pixels, with a maximum of 10 pixels. The error achieved with the Cubic camera was only 0.02 pixels.*

$\mathbf{x} = (x, y, z)^\top$ in space was equal to the x coordinate of the point, and the v coordinate was equal to the radial distance of the point from the line of flight. In symbols

$$\begin{aligned} u &= x \\ v &= \sqrt{y^2 + (z - 3000)^2} \end{aligned}$$

This data was then fitted with a Cubic camera, a Perspective camera and a Linear Pushbroom camera model and the residual reprojection error was recorded. The results are shown in Fig 1.

4 Solving for the Cubic Camera

We now consider the basic photogrammetry problem of parameter estimation for the Cubic camera. We assume given a set of image to world correspondences $\mathbf{u}_i rightarrow \mathbf{x}_i$. The task is to compute the parameters of the Cubic RP camera, namely the coefficients of the polynomials $N_u(\mathbf{x})$, $D_u(\mathbf{x})$, $N_v(\mathbf{x})$ and $D_v(\mathbf{x})$. Two methods will be used to do this.

1. A linear method based on linear least-squares minimization. This method is based on the DLT method ([7]) for estimating the parameters of a projective camera.
2. An iterative method based using the Levenberg-Marquardt method ([6]). The linear method was used to provide an initial estimate of camera parameters, which is refined by iteration. This method was implemented using a general-purpose camera solving program Carmen ([4]). Little more will be said in this report concerning the iterative method.

4.1 Linear Estimation of the Cubic Camera Model.

From the equations

$$\begin{aligned} u &= N_u(\mathbf{x})/D_u(\mathbf{x}) \\ v &= N_v(\mathbf{x})/D_v(\mathbf{x}) \end{aligned}$$

defining the cubic camera model, one may obtain by cross multiplication a pair of equations

$$\begin{aligned} uD_u(\mathbf{x}) - N_u(\mathbf{x}) &= 0 \\ vD_v(\mathbf{x}) - N_v(\mathbf{x}) &= 0 \ . \end{aligned} \tag{2}$$

Although these equations are non-linear in $\mathbf{x}$, they are linear in the coefficients of the polynomials. Since each such correspondence gives a pair of equations, and there are a total of 80 unknown parameters, a total of at least 40 correspondences are required to solve for the polynomial coefficients. With more than 40 points one has an over-determined system of equations which will be solved by least-squares techniques. The total set of equations are of the form $A\mathbf{p} = \mathbf{0}$, where $\mathbf{p}$ is the set of parameters. We are not interested in the trivial solution $\mathbf{p} = \mathbf{0}$. Since the polynomials are homogeneous, their quotient $N_u(\mathbf{x})/D_u(\mathbf{x})$ (and the same thing for v) is independent of scale. We find the parameter vector $\mathbf{p}$ that minimizes $||A\mathbf{p}||$ subject to $||\mathbf{p}|| = 1$. The solution is the singular vector corresponding to the smallest singular value of A ([2]).

This is the barest outline of the method. More will be said later about important implementation details and refinements to this algorithm.

4.2 Degeneracy of the Cubic Model

We would like to be able to treat cameras such as the projective and linear pushbroom cameras as special cases of the Cubic camera and use the same parametrization method for all. Care must be taken in doing this, however because of over-parametrization of the camera model. Consider for instance a set of world to image correspondences $\mathbf{u}_i rightarrow \mathbf{x}_i$ corresponding to a projective camera. In the absence of noise, there will exist linear polynomials $N_u(\mathbf{x})$, $N_v(\mathbf{x})$ and $D(\mathbf{x})$ such that $u_i = N_u(\mathbf{x}_i)/D(\mathbf{x}_i)$ and $v_i = N_v(\mathbf{x}_i)/D(\mathbf{x}_i)$. Unfortunately, these are not the only polynomials which give rise to the correct image points. In particular, one may multiply numerator and denominator by an arbitrary polynomial and obtain the same mapping. In symbols the

value of

$$\mathbf{u}_i = \frac{A(\mathbf{x}_i)N_u(\mathbf{x}_i)}{A(\mathbf{x}_i)D(\mathbf{x}_i)}$$

is constant for all polynomials $A(\mathbf{x}_i)$. Since $N_u(\mathbf{x}_i)$ and $D(\mathbf{x}_i)$ have degree 1 for a perspective camera, $A(\mathbf{x}_i)$ may be an arbitrary degree 2 homogeneous polynomial. Such a polynomial has $C_2^5 = 10$ degrees of freedom. Since the numerator and denominator of $\mathbf{v}_i = N_v(\mathbf{x}_i)/D(\mathbf{x}_i)$ may independently be multiplied by a polynomial $B(\mathbf{x})$, there exists a 20-parameter family of cubic polynomials defining a projective camera mapping. In this case, the matrix A in the set of equations $A\mathbf{p} = 0$ will have diminished rank. In fact A has 80 columns, but its rank will be at most 60 because of the 20-parameter family of solutions. The solution of $A\mathbf{p} = \mathbf{0}$ will not be well defined, and there is no reason to expect the linear solution to be selected, if one is chosen arbitrarily.

In the presence of a degree of noise in the measurements of image points, or 3D points, the matched points $\mathbf{u}_i rightarrow \mathbf{x}_i$ will not correspond precisely with a true perspective model. The cubic model will attempt to correct for this by the introduction of spurious higher-order terms. This will cause the model to match the data more precisely on the measured data. However, it can lead to large errors in other parts of the scene, far from measured control points. In brief, in the presence of instabilities of this nature, one can not extrapolate reliably beyond the measured data. This point will be illustrated later on in this paper.

4.3 Data Normalization

It has been pointed out by several authors, for instance the present authors ([3]) that prenormalization of the input data is essential for obtaining a good result from linear algorithms of this kind which do not minmize geometrically meaningful quantity. Before running the linear algorithm to compute the camera parameters, it is absolutely essential to normalize the data. The general method involves three steps.

1. Choose transforms $T_{\mathbf{u}}$ and $T_{\mathbf{x}}$ of the image and object points such that $\mathbf{u}_i \mapsto \mathbf{u}_i' = T_{\mathbf{u}}\mathbf{u}_i$ and $\mathbf{x}_i \mapsto \mathbf{x}_i' = T_{\mathbf{x}}\mathbf{x}_i$.
2. Solve to find a parametrized Cubic RP camera model, represented by a map P' (in general, nonlinear), that provides a best possible solution to the set of equations $\mathbf{u}_i' = P'\mathbf{x}_i'$. This is done using the linear algorithm described above.
3. Replace P' by $P = T_{\mathbf{u}}P'T_{\mathbf{x}}^{-1}$. This mapping will satisfy $P\mathbf{x}_i = T_{\mathbf{u}}^{-1}P'T_{\mathbf{x}}\mathbf{x}_i = T_{\mathbf{u}}^{-1}P'\mathbf{x}_i' = T_{\mathbf{u}}^{-1}\mathbf{u}_i' = \mathbf{u}_i$ as required. Note that composition of mappings and not matrix multiplication is implied by this juxtaposition $T_{\mathbf{u}}P'T_{\mathbf{x}}^{-1}$, since P' is not linear.

The recommended normalizing transforms $T_{\mathbf{u}}$ and $T_{\mathbf{x}}$ are both of the same type : translation of the data to place its centroid at the origin, and scaling so that the average point is a distance $\sqrt{2}$ from the origin in the case of $T_{\mathbf{u}}$, which is a 2D transformation, and $\sqrt{3}$ in the case of $T_{\mathbf{x}}$, which is a 3D transformation. The reasons for this choice of scaling are given in [3]. The main purpose of data normalization is to improve the conditioning of the problem. To see why this would otherwise be a problem, consider a 3D object point with coordinates $(x, y, z, t)^\top = (500, 500, 500, 1)^\top$ in some coordinate system mapping to an image point $(u, v)^\top = (500, 500)$. In writing the set of equations (2), the entry corresponding to the term x^3 of N_u will be 500^4, whereas the entry corresponding to term t^3 of D_u will be 1. This wide range of entries in matrix A means that A will be poorly conditioned, and the solution very unstable in the presence of noise. The normalization transformations are designed to give each entry in A an equal weight.

In doing this, it is important that if P' is a cubic RP mapping, then so is $P = T_{\mathbf{u}}^{-1}P'T_{\mathbf{x}}$. This will be true for any linear transformation $T_{\mathbf{x}}$, but interestingly enough, not for any $T_{\mathbf{u}}$. This is easily seen as follows. First, suppose that $P_u'(\mathbf{x}) = N_u'(\mathbf{x})/D_u'(\mathbf{x})$ where both N_u' and D_u' have degree n. (For the Cubic camera, $n = 3$.) Then, $P_u'T_{\mathbf{x}}(\mathbf{x}) = N_u'(T_{\mathbf{x}}(\mathbf{x}))/D_u'(T_{\mathbf{x}}(\mathbf{x}))$, and both numerator and denominator are degree n polynomials in $\mathbf{x}$, since $T_{\mathbf{x}}(\mathbf{x})$ is linear.

On the other hand, consider $TP'(\mathbf{x})$ where T is any affine transformation. The u coordinate in the image is given by $u = T(u', v') = \alpha u' + \beta v' + \gamma$. The v coordinate is expressed similarly. In this case, we have

$$TP'(\mathbf{x}) = \alpha\frac{N_u'(\mathbf{x})}{D_u'(\mathbf{x})} + \beta\frac{N_v'(\mathbf{x})}{D_v'(\mathbf{x})} + \gamma$$

$$= \frac{\alpha N_u'(\mathbf{x})D_v'(\mathbf{x}) + \beta N_v'(\mathbf{x})D_u'(\mathbf{x}) + \gamma D_u'(\mathbf{x})D_v'(\mathbf{x})}{D_u'(\mathbf{x})D_v'(\mathbf{x})}$$

Thus, the degree of the RP transformation is increased. There are two evident exceptions to this :

1. $D_u' = D_v'$. This is the case for a projective camera.
2. $\beta = 0$. This is the case in which the transformation is a simple scaling and translation. This is the recommended sort of transformation.

In this latter case, one has

$$TP'(\mathbf{x}) = \frac{\alpha N_u'(\mathbf{x}) + \gamma D_u'(\mathbf{x})D_v'(\mathbf{x})}{D_u'(\mathbf{x})} \tag{3}$$

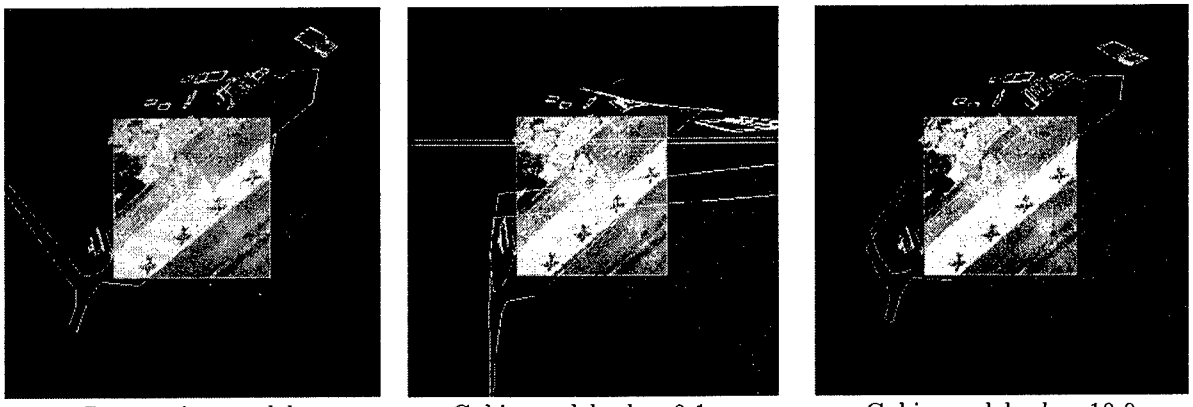
4.4 Computing Composition of Mappings

The composition of $T_{\mathbf{u}}^{-1}P'(\mathbf{x})$ is easily computed using (3). The computation of $P'T_{\mathbf{x}}$ is a little more complicated, however. This is a standard sort of algebraic manipulation problem, but complicated if one does not do it the right way. A simple implementation is possible using tensors, as described now.

A cubic homogeneous polnomial $N(\mathbf{x})$ may be written in terms of a symmetric tensor N_{ijk} defined such that if $\mathbf{x}$ has components x^i, then $N(\mathbf{x}) = N_{ijk}x^ix^jx^k$. Here, the superscripts represent indices, not powers, and a repeated index in the upper and lower positions implies summation. This may be more familiar in the degree 2 case, where a quadratic form may be written as $\mathbf{x}^\top A\mathbf{x} = A_{ij}x^ix^j$.

Now, if we apply a linear transformation such that $x^i = T_p^i x'^p$, then it follows that $N(\mathbf{x}) = N_{ijk}x^ix^jx^k = N_{ijk}T_p^iT_q^jT_r^kx'^px'^qx'^r = N'_{pqr}x'^px'^qx'^r$ where N'_{pqr} is defined by

$$N'_{pqr} = N_{ijk}T_p^iT_q^jT_r^k \tag{4}$$



Perspective model | Cubic model : $k = 0.1$ | Cubic model : $k = 10.0$

Figure 2: *This shows the result of camera resectioning using the Cubic camera model. Hand-picked correspondences between a site model and an image are used to compute the camera model. The site model is then projected into the image using the computed camera model. The three examples show a projective camera, and two parametrized cubic cameras computed with different settings of k, the constraint weight for high-order coefficients. In all cases, the site model is well aligned with the image within the image area. For the cubic camera, the agreement of the site model outside of the area where control points are chosen is not so good, though for larger value of k, the site model is projected reasonably well.*

This is the composition rule required to compute the composition $PT_{\mathbf{x}}$.

4.5 Regularization

It was shown in a previous section that in cases where a Cubic camera is well approximated by a projective camera, or some other linear camera, the computation of the camera model may be unstable. A way that we have found useful for dealing with this problem is regularization. In this method, a constraint is put on quadratic and cubic terms in N_u, N_v, D_u and D_v constraining them to be close to zero. This constraint is weighted by a parameter k, where high values of k provide a strong constraint on the values of the higher order terms. The requirements that these terms be small is balanced against the requirements imposed by the data.

This has two effects :

1. Low degree polynomials are favoured over high-degree polynomials. This has the effect of resolving the ambiguity in the set of solutions. In the presence of only a low degree of noise, for instance, a perspective cameras will be modelled with linear (or near linear) rational polynomials.
2. It is possible to solve for the camera parameters with fewer than the full number (40 in the Cubic camera case) of point correspondences. This is useful when it is difficult to find such a large number of control points.

Figure 2 shows the effect of different values of k.

References

[1] Eamon B. Barrett, Paul M. Payton, and Joseph L. Mundy. Focus : A shared vision technology transfer project. In *Proc. DARPA Image Understanding Workshop*, 1997.

[2] O. Faugeras. *Three Dimensional Computer Vision: A Geometric Viewpoint.* The MIT Press, Cambridge, MA, 1993.

[3] R. I. Hartley. A linear method for reconstruction from lies and points. In *Proc. International Conference on Computer Vision*, pages 882 – 887, 1995.

[4] Richard I. Hartley. An object-oriented approach to scene reconstruction. In *Proc. IEEE International Conference on Systems Man and Cybernetics, Peking*, pages 2475 – 2480, October 1996.

[5] Richard I. Hartley and Rajiv Gupta. Linear pushbroom cameras. In *Computer Vision - ECCV '94, Volume I, LNCS-Series Vol. 800, Springer-Verlag*, pages 555–566, May 1994.

[6] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1988.

[7] I.E. Sutherland. Three dimensional data input by tablet. *Proceedings of IEEE*, Vol. 62, No. 4:453–461, April 1974.

Rosetta: An Image Database Retrieval System

Jeremy S. De Bonet and Paul Viola*
Learning & Vision Group
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
E-MAIL: jsd@ai.mit.edu , viola@ai.mit.edu
HOMEPAGE: http://www.ai.mit.edu/projects/lv

Abstract

A new algorithm is presented which approximates the perceived visual similarity between images. The images are initially transformed into a feature space which captures visual structure, texture and color using a tree of filters. Similarity is then measured as the distance in this *perceptual feature space*. Using this algorithm we have constructed an image database system, called Rosetta, which can perform example based retrieval on large image databases. A typical query consists of a small set of images which are representative of a broader class (e.g. images of automobiles or images of city skylines). From the example images a characteristic signature in feature space is computed and is compared to the features of each image in the database. The closest database images are returned. Performance in this area is notoriously difficult to quantify. We have acquired a set of 2900 images which have been divided into 29 classes based on visual and semantic similarity. Using a small set of randomly selected images from each class as a query we can with high reliability return other images from that class and reject images from other classes.

1 Introduction

There are many potential applications for a content based image database retrieval program. For example users may wish to search through all television news broadcasts from Moldova for images of military vehicles; or to search through the World Wide Web for images of the Eiffle tower. Today such queries must be performed manually. Though people can perform searches for complex or loosely defined images – finding images involving political violence, or depicting "pride" – they typically must examine each image in the database. The size of many image databases, however, have grown beyond the scope of manual searching. Clearly some scheme for automatically and efficiently searching very large image databases is necessary. Such a system must be able to measure and compare key visual components of natural images.

A digitized image can be interpreted as a single very high dimensional point in pixel space. From this point of view, it is not unreasonable to consider the distance between images in pixel space as a measure of the visual similarity between images. Clearly if two images are very near in pixel space they look similar. Unfortunately images which are far apart in pixel space are often very similar in visual content. What is needed is some sort of "Rosetta stone" which can translate images into another representation which would allow us to interpret and compare them based on their content and visual structure.

Many algorithms have been proposed for image database retrieval. For the most part these techniques compute a feature vector from an image which is made up of a handful of image measurements. Visual or semantic distance is then equated with feature distance. Examples include color histograms, texture histograms, shape boundary descriptors, eigenimages, and hybrid schemes [QBIC, , Niblack *et al.*, 1993, Virage, , Kelly *et al.*, 1995, Pentland *et al.*, 1995, Picard and Kabir, 1993, Santini and Jain, 1996]. A query to such a system typically consists of specifying two types of parameters: the target values of each of the measurements; and a set of weights, which determine the relative importance of deviations from the target in each measurement dimension. The features used by these systems each capture some non-specific property of images. As a result of their generality however, many images which are actually very different in content, generate the same feature responses. In many images the critical structural properties that determine the content of the image cannot be dis-

*Research supported in part by DARPA under ONR contract No. N00014-95-1-0600 and by the Office of Naval Research under contract No. N00014-96-1-0311. Portions of this work were done at the Microsoft Vision Technology group, Microsoft Inc. Redmond, Wa.

criminated by these highly-general features. For example features based on color histograms would easily confuse a photo of a stack of white papers with a photo taken outdoors in the snow. Recently a novel approach which does not use feature vectors has demonstrated very strong results [Lipson *et al.*, 1997]. Their system measures the global structure images and is relatively insensitive to local image texture.

The goal of our approach is to pay attention both to local texture and global structure. We hypothesize that there is in fact no clear distinction between local texture and global structure: that they are simply two ends of a continuum. Our algorithm represents images at many levels of resolution: measuring color, edge orientation, and other local properties at each resolution. The visual properties captured by these local operations changes at different scales. A horizontal color edge at a high resolution might be related to the leaves of a tree, while a horizontal color edge at a much lower resolution might be caused by a blue sky above a green field. This sort of multi-scale feature analysis is of critical importance. It has been used successfully in the context of object recognition [Rao and Ballard, 1995, Viola, 1996].

Our system differs from others because it detects not only first order relationships, such as the edges described above, but also measures how these first order relationships are related to one another. Thus by finding patterns between image regions with particular local structural organization, more complex – and therefore more discriminating – features can be extracted. In essence the Roesetta system looks for textures of textures. For example, at the highest level of resolution, vertical edge detectors will respond both to skyscrapers and picket fences. At this resolution the two images are not distinguished by the presence of vertical texture. If we examine the spatial organization of the vertical texture we find that picket fences yield horizontal bars of vertical energy. It is the non-linear conjunction of texture and spatial organization that allows our system to distinguish a variety of complex images.

There are tens of potentially useful color and texture features which occur in local regions of natural images. There are hundreds of conjunctive features that can be formed by computing a feature at one resolution and then measuring its structural organization at another. This analysis can be repeated many times: in effect yielding measures of higher order textural organization. There are literally thousands of these multiply conjoined features. Taken together such a representation is called the characteristic signature of an image.

No human being can be expected to determine desired values or weights for so many conjunctive features. Instead, a user retrieves images from the Rosetta system by presenting a set of example images. The system computes the desired feature values and weights from this set. Thus, this paradigm can be described as "query by image example." Variations in the feature vectors of the query images are used to determine the relative importance of each image feature in the query. Those features which have consistent values across all the example images receive the largest weights. Weighting in this way causes those features which are consistent within a class to be most important in determining class membership. For example, in one query chromatic-content may be the primary measure, while in another, spatial-arrangement may be dominant.

2 Computing the Characteristic Signature

The "texture-of-texture" measurements used by the Rosetta system are based on the outputs of a tree of non-linear filtering operations. Each path through the tree creates a particular filter network, which responds to certain structural organization in the image. Measuring the appropriately weighted difference between the signatures of images in the database and the set of query-images, produces a similarity measure which can be used to rank and sort the images in the database.

The computation of the characteristic signature is straightforward. At the highest level of resolution the image is convolved with a set of local linear features. In our experiments there are 25 local features including oriented edges and bars. The results of these convolutions are 25 feature response images. These images are then rectified by squaring, which extracts the *feature energy* in the image, and then downsampled by a factor of two. This convolution, rectification and downsampling is then repeated on each of these 25 half resolution images producing 525 quarter scale texture-of-texture energy images. Repeating this procedure a third time yields 15,625 meta-texture feature images at eighth scale. The sum of the values in each of these images provides one element in the characteristic signature. This is done independently for each color channel in the input image, creating a signature which contains 46,875 measurements. By exploiting the hierarchical construction of these measurements, the characteristic signature for an image can be computed in about a minute on a workstation. Once computed, this signature is saved and reused for subsequent queries made on the database.

More formally the characteristic signature of an im-

age is given by:

$$S_{i,j,k,c}(I) = \sum_{pixels} E_{i,j,k}(I_c) \quad (1)$$

where I is the image, i, j and k index over the different types of linear filters, and I_c are the different color channels of the image. The definition of E is:

$$E_i(I) = 2 \downarrow [(F_i \otimes I)^2] \quad (2)$$

$$E_{i,j}(I) = 2 \downarrow [(F_j \otimes E_i(I))^2] \quad (3)$$

$$E_{i,j,k}(I) = 2 \downarrow [(F_k \otimes E_{i,j}(I))^2] \, . \quad (4)$$

where F_i is the ith filter and $2 \downarrow$ is the downsampling operation.

3 Using Characteristic Signatures To Form Image Queries

In our image query paradigm, we describe similarity in terms of the difference between a database-image and a group of example query-images. This is done by comparing the characteristic signature of each image in the database to the mean signature of the query-images. The relative importance of each element of the characteristic signature in determining similarity is proportional to the inverse variance of that element across the example-image group. This has the effect of normalizing the vector-space defined by the characteristic signatures, so that characteristic elements which are salient within the group of example-images contribute more to the overall similarity of an image.

The similarity between an image and the group of query-images is the negative of the sum squared difference between the average query-image signature and the database-image signature weighted by the variance of across the query-image signatures:

$$L = -\sum_i \sum_j \sum_k \sum_c \frac{\left[\overline{S_{i,j,k,c}(I_q)} - S_{i,j,k,c}(I_{test})\right]^2}{Var\left[S_{i,j,k,c}(I_q)\right]} \quad (5)$$

4 Experiments

An image database query system must retrieve images which are similar to those for which the user is searching. Because the concept of *similarity* in the goal above is not well defined it is difficult to quantify query results.

We used a database of 2900 images from 29 Corel Photo CD (collections 1000-2900.) Each CD contains 100 images which have been categorized by theme. Examples of these themes include "Sunsets & Sunrises," and "Mountains of America," as well as less specific collections such as "Spirit of Buddha," or "Christmas Collection," and classes which contain images which are very similar, i.e. "Exotic Cars," and "Auto Racing." Each image has been placed exclusively into one category, however, some could reasonably belong in multiple categories. For example, consider categorizing an image depicting a sunrise over the Rockies, or a 1967 Porsche. Because of this lack of mutual exclusion between true category membership, we would not expect any image query system – or human – to exactly select the same images for a category as did the original classifier.

Figures 1 and 2 show the results of typical user query on this system. The images in Figure 1 are the query-images submitted by the user. In Figure 2 are the thirty images found to be most similar; similarity decreases from upper left (most similar) to lower right. Though these examples provide an anecdotal indication that the system is generating similarity measures which roughly conform to human perception, it is difficult to ascertain from them a quantitative evaluation of the system. To better measure the performance of the system two experiments on this database were performed.

5 Experiment 1

The images in each collection were selected because they were determined by a human observer to be representative examples of the theme of the collection. Threfore it is a vaild experiment to take a few images from a category and use them to retrieve other images from that category. The ranking of the true images in that category provide a measure of success.

The results of two queries are shown in the receiver operating characteristics curve in figure 3.

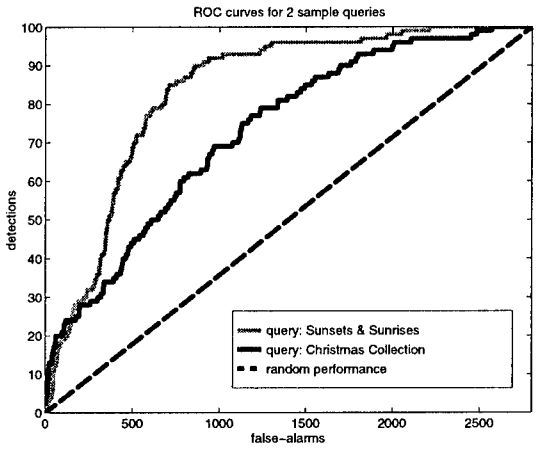


Figure 3: The ROC curve for two queries.

For each query four images were randomly chosen from a single image collection. The similarity was then measured between this query set and all the images in the database. The number of images from

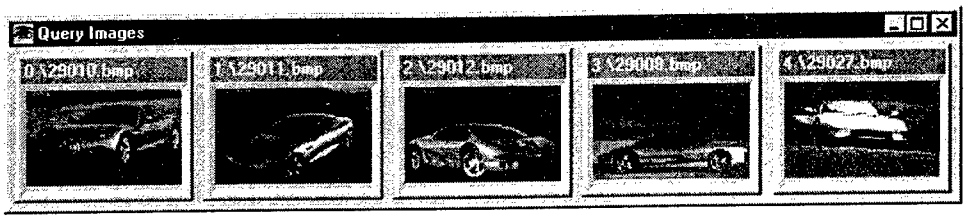


Figure 1: A sample query intended to return images of cars.

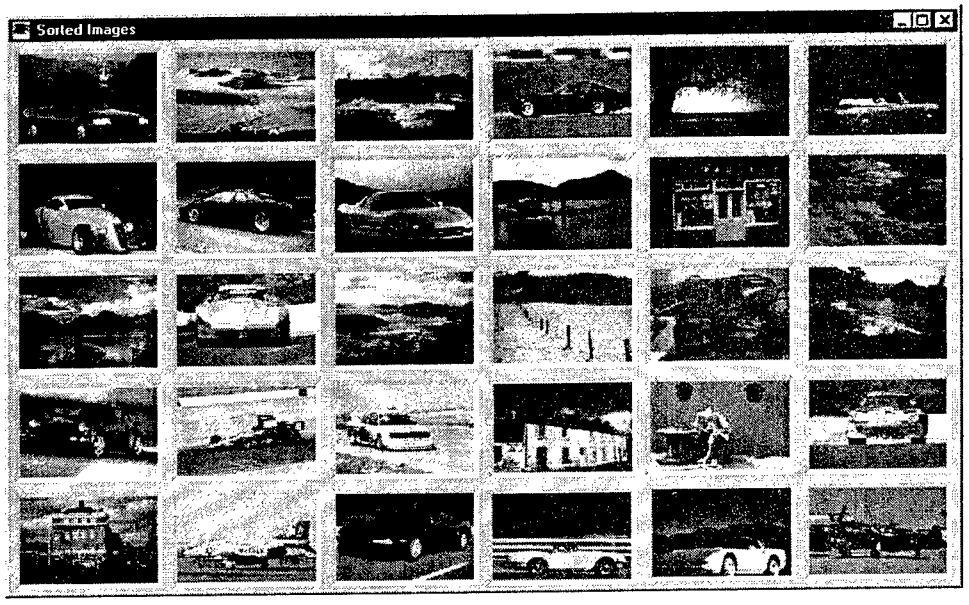


Figure 2: The response of the Rosetta system to the query in Figure 1.

the target collection is plotted against the number of other images as a function of image similarity.

The top (grey) curve was generated by a query for images from the "Sunsets & Sunrises" collection, which contains images which all share common visual characteristics. Two of the images from this collection are shown in figure 4b. Though there is significant chromatic variation between the various images in this group they all share very similar structural characteristics.

The middle (black) curve in Figure 4b, shows a query from the "Christmas collection," whose images contain far more visual variety. Two typical images from this collection are shown in Figure 5b. Because of this increased variety one would anticipate poorer performance for such a query. However, performance is still significantly better than chance, which is indicated by the diagonal dashed line. This is due to the fact that though there is significant variety, many of the images do still contain similar structures.

6 Experiment 2

A second measure of the Rosetta system's performance is obtained by measuring how well the system can discriminate between two classes given a few examples from each.

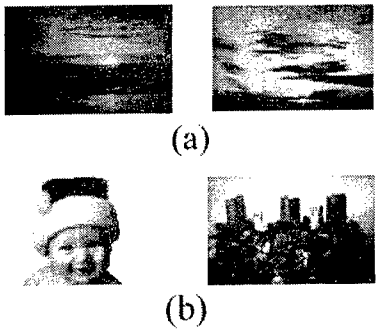


Figure 4: Example images from two categories in the database.

Using examples from only the "Sunsets & Sunrises" collection, and attempting to classify images from both that collection and from the "Christmas Collection" the black curve in Figure 5 is obtained. On this plot, the number of correctly classified images (number of detections) is plotted against the number incorrectly classified (false-alarms.) From this curve we can see, for example, that if searching for "Sunsets & Sunrises", about of 32 out of the top 50

responses would be correctly classified yielding an accuracy of only 64 percent. Chance, represented by the dashed line is 50 percent. If however, we present the Rosetta system with examples of each collection, the grey curve in Figure 5 is obtained. On this curve 43 out of the top 50 are correctly classified, yielding 86 percent accuracy. Because of the large variation within the image collections, as discussed above, performance drops off steadily as we consider retrieving successively larger portions of the target collection.

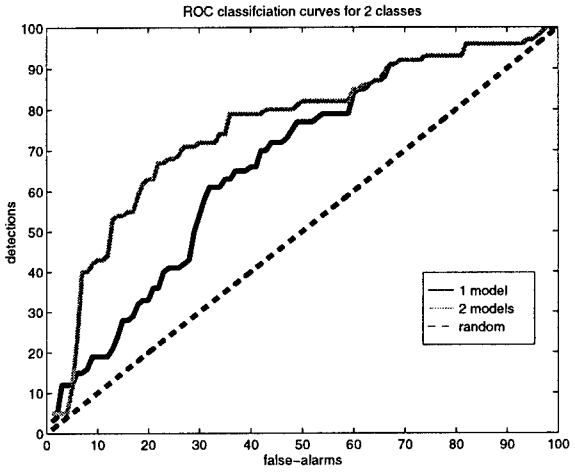


Figure 5: Using examples from multiple classes (grey curve) improves performance over examples from just one class (black curve.)

7 Discussion

We have presented a technique for approximating perceived visual similarity, by measuring the structural content similarity between images. We have developed a system called Rosetta which transforms images into a very high dimensional "characteristic signature" space which captures the visual structure in the image. Using this represntation, the Rosetta system directly compares database-images to a set of query-images.

Experiments indicate that the Rosetta system can retrieve images which share visual characteristics with the query-images, from a large non-homogeneous database. Because the characteristic signature space incorporates structural information, it can perform queries where simpler methods, such as color histogramming fail. Though the results of queries using the Rosetta system are encouraging, they are not perfect – as evidenced by the false alarms in Figure 2 – we believe that with additional research its performace will improve.

In experiment 2 we demonstrated retrieval performance can be improved by modelling distracting images as a spearate class.

Acknowledgments

The authors wish to thank the Microsoft Vision Technology group for the use of is Vision Software Development Kit (MSVisSDK).

References

[Kelly *et al.*, 1995] M. Kelly, T. M. Cannon, and D. R. Hush. Query by image example: the candid approach. *SPIE Vol. 2420 Storage and Retrieval for Image and Video Databases III*, pages 238–248, 1995.

[Lipson *et al.*, 1997] P Lipson, E Grimson, and P Sinha. Configuration based scene classification and image indexing. In *Computer Vision and Pattern Recognition*, 1997.

[Niblack *et al.*, 1993] V. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The qbic project: querying images by content using color, texture, and shape. *IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science & Technology*, 1908:173–187, 1993.

[Pentland *et al.*, 1995] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. Technical Report 255, MIT Media Lab, 1995.

[Picard and Kabir, 1993] R. W. Picard and T. Kabir. Finding similar patterns in large image databases. *ICASSP*, V:161–164, 1993.

[QBIC,] QBIC. The ibm qbic project. Web: http://wwwqbic.almaden.ibm.com/.

[Rao and Ballard, 1995] R. P. N. Rao and D.H. Ballard. Object indexing using an iconic sparse distributed memory. Technical Report TR-559, University of Rochester, 1995.

[Santini and Jain, 1996] S Santini and R Jain. Gabor space and the development of preattentive similarity. In *Proceedings of ICPR 96*. International Conference on Pattern Recognition, Vienna, August 1996.

[Viola, 1996] Paul Viola. Complex feature recognition: A bayesian approach for learning to recognize objects. Technical Report 1591, MIT AI Lab, 1996.

[Virage,] Virage. The virage project. Web: http://www.virage.com/.

The Earth Mover's Distance, Multi-Dimensional Scaling, and Color-Based Image Retrieval

Yossi Rubner, Leonidas Guibas, Carlo Tomasi *
Computer Science Department, Stanford University
Stanford, CA 94305
[rubner,guibas,tomasi]@cs.stanford.edu

Abstract

In this paper we present a novel approach to the problem of navigating through a database of color images. We consider the images as points in a metric space in which we wish to move around so as to locate image neighborhoods of interest, based on color information. The data base images are mapped to distributions in color space, these distributions are appropriately compressed, and then the distances between all pairs I, J of images are computed based on the work needed to rearrange the mass in the compressed distribution representing I to that of J. We also propose the use of multi-dimensional scaling (MDS) techniques to embed a group of images as points in a two- or three-dimensional Euclidean space so that their distances are preserved as much as possible. Such geometric embeddings allow the user to perceive the dominant axes of variation in the displayed image group. In particular, displays of 2-d MDS embeddings can be used to organize and refine the results of a nearest-neighbor query in a perceptually intuitive way. By iterating this process, the user is able to quickly navigate to the portion of the image space of interest.

1 Introduction

Rummaging through a large catalog of pictures in search of a particular image is unrewarding and time-consuming. Image database retrieval research [Bach *et al.*, 1996; Guibas and Tomasi, 1996; Niblack *et al.*, 1993; Pentland *et al.*, 1996] attempts to automate parts of this task. The most popular proposals for formulating a query into an image database is to sketch the desired picture or to provide an example of a similar image. Yet often we do not know the precise appearance of the desired image(s). We may want a sunset, but we do not know if sunsets in the database are on beaches or against a city skyline. When looking for unknown images, browsing, not query, is the preferred search mode. And the key requirement for browsing is that similar images are located nearby. Current retrieval systems list output images in order of increasing distance from the query. However, the distances among the returned images also convey useful information during browsing. In this paper, we present a novel framework for computing the distance between images, and a set of tools to visualize an entire image data base or parts of it during browsing.

The question of image similarity is complex and delicate. Semantic similarity (two images with cats are similar to each other) is still out of the question, and we must make do with similarity of appearance. More specifically, in this paper we focus on the overall color content of an image as the main criterion for similarity. The overall distribution of colors within an image contributes to the mood of the image in an important way, and is a useful clue for the image's contents. Sunny mountain landscapes, sunsets, cities, faces, jungles, candy, and fire fighters scenes lead to images that have different but characteristic color distributions. If the pictures in a database can be arranged in a geometric space so that their locations reflect differences and similarities in their color distributions, browsing the database becomes intuitively meaningful. In fact, the database is now endowed with a metric structure, and can be explored with a sense of continuity and comprehensiveness: all we care, as far as the parts of the database that have undesired color distributions are concerned, is that we need not traverse them. On the other hand, interesting regions can be explored with a sense of getting closer

*This work was sponsored by the Defense Advanced Research Projects Agency under contract DAAH04-94-G-0284 monitored by the US Army Research Office. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the United States Government, or Stanford University.

or farther away from the desired distribution of colors. In summary, the user can form a mental, low-detail picture of the entire database, and a more detailed picture of the more interesting parts of it. If a picture is worth a thousand words, a picture of an image database is worth a whole book.

Of course, arrangement criteria other than color distribution are possible. For instance, information about the position of colors in the images, as well as shape and texture, ought to be considered eventually. However, color distribution is at the same time useful in its own right and complex enough to let us illustrate the main issues. Thus, while we experiment with the notion of similarity in the context of color information, we define a framework in which shape and texture descriptors can also be accommodated, leading to a skeletal theory of image database visualization. In particular, we address the following questions:

- How do we summarize the color distribution of an image?
- When do two images have similar color distributions and, more generally, how do we measure the 'distance' between these distributions?
- How can we arrange a collection of images so that similar images are near each other?

Summarization of color distribution has to do with perceptual significance, invariance, and efficiency at the same time. Colors should be represented in a way that reflects a human's appreciation of similarities and differences. At the same time, the distribution of colors in an image should be represented by a collection of data that is small, for efficiency, but rich enough to reproduce the essential information. The issue of relevance to human perception has been resolved by the definition of appropriate color representations, among which we choose the CIE-LAB standard. Section 2 addresses the issue of summarization by presenting a new, efficient clustering scheme based on k-d trees. This scheme buys efficiency at the expense of reduced guarantees about the size of the output. While more expensive algorithms may guarantee a minimal number of clusters, this is an unnecessary requirement for our application. The result of this method is a small collection of (weighted) points in color space which represent well the full distribution; we call this set of points a (color) *signature*. Section 3 introduces the *Earth Mover's Distance* (EMD) [Stolfi, 1994] as a useful and flexible measure of distance between signatures, and presents an efficient algorithm for its computation based on linear programming. This distance endows the image database with an appropriate metric, thereby addressing the question of image similarity. Section 4 addresses the third question above, and shows how to use the technique of *Multi-Dimensional Scaling* (MDS) [Kruskal, 1964] in order to visualize either the entire database or just the part of it returned in response to a query in a two- or three-dimensional space. The resulting composite image properly reflects the distribution of color distributions within the database. Finally, section 5 argues that the techniques and issues introduced in this paper generalize to other aspects of image description.

2 Color Signatures

The color information of each image is reduced to a compact representation that we call the *signature* of the image. In general a signature contains a varying number of points in a Euclidean space where a weight is attached to each point. In the case of color images, the points represent clusters of similar colors and the weight of a point is the fraction of the image area with that color.

To compute the signature of a color image, we first slightly smooth each band of the image's RGB representation in order to reduce possible color quantization and dithering artifacts. We then transform the image into the CIE-LAB color space [Wyszecki and Styles, 1982] using D65 as the reference white. This nonlinear transformation deforms the RGB color space so that the resulting Euclidean distance between color coordinates approximates how well colors are discriminated by humans.

Each image implies a distribution of points in the three-dimensional CIE-LAB color space where a point corresponds to a pixel in the image. We coalesce this distribution into clusters of similar colors. We define these as clusters that do not exceed 30 units in any of the L, a, b axes. Because of the large number of images to be processed in a typical database, clustering must be performed efficiently. To this end, we devised a novel two-stage algorithm based on a k-d tree [Bentley, 1975]. In the first phase, approximate clusters are found by a balanced partition of color space through a k-d tree. Subdivision stops when a cell becomes smaller than the allowed cluster size. This process can result in excessive subdivision. The second phase then tries to merge close clusters computed in the first phase by performing a second k-d tree clustering on points which represent the centroids of

the clusters that are produced in the first phase, after shifting the space coordinates by one half of the minimal allowed cell size. Each cluster contributes a pair (p, w_p) to the signature representation of the image where p is the average color of the cluster and w_p is its weight which is the fraction of image pixels that are in that cluster. Figure 2 shows examples of color signatures for three images.

The signatures thus obtained are compact: the color distribution of an entire image is summarized by a handful of points, typically eight to twelve. Because of the clustering algorithm used, signatures represent well the image's overall color distribution. Since signatures represent distributions in the CIE-LAB color space, they are perceptually significant, in that Euclidean distances between points are strongly correlated with perceptual differences. Because of clustering, small variations in the colors of an image have little effect on signatures, thereby providing a moderate degree of invariance to changes of viewpoint and lighting. Finally, signatures are simple and flexible abstractions for which we can define meaningful metrics, as shown in the following section.

3 Distance Between Color Signatures

In image retrieval, it is important to define a similarity measure between two color distributions or, in particular, between two color signatures. When considering only the color content of images, and ignoring the actual positions of the pixels within the image, this problem is known as the color indexing problem which was introduced by Swain and Ballard [Swain and Ballard, 1991] and was approached in several ways by others [Hafner *et al.*, 1995; Stricker and Orengo, 1995; Werman *et al.*, 1985]. Our approach is closest to, but more general and at the same time more efficient than that of [Werman *et al.*, 1985]. The other methods are bound to retrieve false positives [Stricker and Orengo, 1995]. We define the distance between two signatures to be the minimum amount of 'work' needed to transform one signature into the other (figure 1). The work needed to move a point, or a fraction of a point, to a new location is the portion of the weight being moved, multiplied by the Euclidean distance between the old and the new locations. When changing one signature to another, the work is the sum of the work done by moving the weights of the individual points of the source signature to those of the destination signature.

We allow the weight of a single source signature point to be partitioned among several destination signature points, and vice versa. We call this distance function the *earth mover's distance*. This is a name suggested by Stolfi [Stolfi, 1994], by analogy with some CAD programs for road design which have a function that computes the optimum earth displacement from roadcuts to roadfills. As compared with the match distance of [Werman *et al.*, 1985], our distance is more general because it allows fractional/partial matches. Furthermore, it can be computed much more efficiently, as we now show.

The earth mover's distance computation can be formalized as the following linear programming problem: Given two signatures: $\mathbf{p} = \{(p_1, w_{p_1}), \ldots, (p_m, w_{p_m})\}$ and $\mathbf{q} = \{(q_1, w_{q_1}), \ldots, (q_n, w_{q_n})\}$ where p_i and q_j are points in some Euclidean space, the CIE-LAB color space in our case, and w_{p_i}, w_{q_j} are the corresponding weights of the points, find an $m \times n$ cost matrix $\mathbf{C}$ where C_{ij} is the amount of weight op p_i matched to q_j, that will minimize the function:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}\,\|p_i - q_j\|$$

($\|\cdot\|$ is the Euclidean distance) subject to the following constraints:

$$C_{ij} \geq 0 \qquad 1 \leq i \leq m\,,\ 1 \leq j \leq n \tag{1}$$

$$\sum_{i=1}^{m} C_{ij} \leq w_{q_j} \qquad 1 \leq j \leq n \tag{2}$$

$$\sum_{j=1}^{n} C_{ij} \leq w_{p_i} \qquad 1 \leq i \leq m \tag{3}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij} = \min(w_{\mathbf{p}}, w_{\mathbf{q}})\,. \tag{4}$$

where $w_{\mathbf{p}} = \sum_{i=1}^{m} w_{p_i}$ and $w_{\mathbf{q}} = \sum_{j=1}^{n} w_{q_i}$. The earth mover's distance is defined as the normalized distance between points $\mathbf{p}$ and $\mathbf{q}$:

$$\begin{aligned}\mathrm{EMD}(\mathbf{p},\mathbf{q}) &= \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}\,\|p_i - q_j\|}{\sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}} \\ &= \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}\,\|p_i - q_j\|}{\min(w_{\mathbf{p}}, w_{\mathbf{q}})}\,.\end{aligned}$$

Constraint 1 allows only for positive amounts of 'earth' to be moved. Constraints 2 and 3 limit the capacity of 'earth' a point can contribute to the weight of the point. Constraint 4 forces at least one of the signatures to use all of its capacity, otherwise a trivial solution is not to move any 'earth' at all.

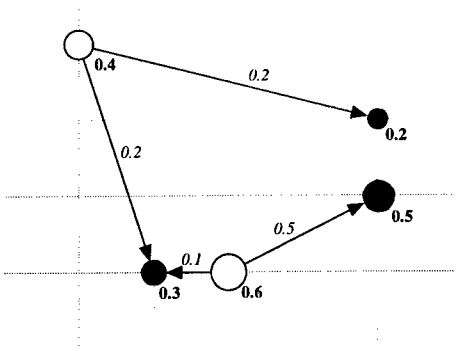


Figure 1: *The earth mover's distance in 2D between a signature with three points (black) and one with two (white). Bold and italic numbers are the weights of the points and the weights moved between points, respectively.*

The earth mover's distance has many desirable properties relevant to our application. As long as the total weight of each of our signatures is the same, the earth mover's distance is symmetric and satisfies the triangle inequality — thus we really work with a metric space. The 'optimal assignment' problem which the earth mover's distance computes also gives us a way to 'morph', or continuously transform, two distributions into each other: simply imagine the appropriate weight fractions moving at constant rates along the segments joining the corresponding source and destination points in color space. During the morph the centroid of the morphing distribution will move continuously from the centroid of the source to that of the destination signature. This shows that the distance between the centroids of the two signatures involved, is a lower bound on the earth mover's distance: Assuming that $w_{\mathbf{p}} = w_{\mathbf{q}} = w$ then

$$\mathrm{EMD}(\mathbf{p}, \mathbf{q}) \geq \|\mathbf{p}_{ave} - \mathbf{q}_{ave}\|$$

where

$$\mathbf{p}_{ave} = \frac{1}{w} \sum_{i=1}^{m} w_{p_i} p_i \qquad \mathbf{q}_{ave} = \frac{1}{w} \sum_{j=1}^{n} w_{q_j} q_j \ .$$

This is useful for quickly recognizing dissimilar distributions.

Notice that in our formulation we do allow the total weights of the two signatures to be different. This is useful for content-based image retrieval systems for example, when a color query specifies only a part of the wanted color distribution, leaving the rest as "don't care". In this case, of course, the EMD is not a true distance and the lower bound we show does not hold.

The earth mover's distances between the images in figure 2 can be summarized by the following symmetric distance matrix:

$$\begin{bmatrix} 0 & 19.46 & 71.94 \\ 19.46 & 0 & 60.03 \\ 71.94 & 60.03 & 0 \end{bmatrix} .$$

As expected, the first two images are relatively close since they contain similar colors (blues and greens). The third image is relatively far from the first two but somewhat closer to the second image because the colors of the house and the trees in the second image are similar to the colors of the sunset in the third image.

4 Database Visualization

A metric for color signatures is crucial for image retrieval, because it quantifies the intuitive notion of image similarity. If the metric corresponds to perceptual similarity, retrieving images in response to a given query amounts to returning images whose distance from the query is small in the space of color signatures. While the earth mover's distance is indeed at the core of our image retrieval system, and has proven very effective, in this paper we want to emphasize a related but distinct use of the signature metric defined in the previous section. When browsing an image database, we often have only a vague idea of what our target images look like. This is especially true when we have not seen the images in the database beforehand. The standard format of interaction with the database, that is, iterations of a query answered by the presentation of a list of images, is not satisfactory in this case. First, one would like to have a global view of the returned images. As figure 3 (a) shows, images in the returned list can be related to one another and yet appear at separate places in the list. The returned images should be displayed not only in order of their distance from the query, but also arranged according to their mutual distances. In brief, the user of the system would benefit from a more coherent view of the query results.

Second, browsing and navigating in a large database is disorienting unless the user can form a mental picture of the entire database. Only having an idea of the surroundings can offer an indication of where to go next. The wider the horizon, the more secure navigation will be. How can such a global picture of an image database be created? Signatures offer once again a solution. Our earth mover's distance quantifies the perceptual difference that separates two signatures. Consequently, each signature can be represented by a single point in a suitably high-dimensional space, such that distances between these points are equal to

the earth mover's distances between the corresponding signatures. The computation of the coordinates of these high-dimensional points is called an *embedding*. However, humans can only visualize low-dimensional spaces, typically in two or three dimensions. We then look for an approximate embedding, rather than for an exact one.

The approximate embedding problem was formalized by Kruskal [Kruskal, 1964] into the so-called Multi-Dimensional Scaling (MDS) problem. Given a set of n objects together with the matrix of distances δ_{ij} between them, and given a (small) dimension d, the problem is to find a set of n points in d-dimensional space whose distances $\{\hat{\delta}_{ij}\}$ are as close as possible to the original distances $\{\delta_{ij}\}$. The choice of closeness that was suggested by Kruskal is to minimize:

$$\text{STRESS} = \left[\frac{\sum_{i,j}(\hat{\delta}_{ij} - \delta_{ij})^2}{\sum_{i,j}\delta_{ij}^2}\right]^{1/2}.$$

Rigid transformations and reflections can be applied to the MDS result without changing the STRESS. Using MDS can assist navigation in the space of images both locally and globally, as we now illustrate.

4.1 Local MDS

Performing MDS on the images returned from a query gives us a better way to display the query results. Instead of the traditional one-dimensional list of images sorted by their distances from the query, we can display a two or three dimensional map of the images, where each image is positioned according to the MDS result. In this way we are presenting information reflecting $\binom{n}{2}$ distances, instead of only n in the traditional method. In addition to visually representing the relative distances between *all* pairs of images, images with similar color content tend to group together. Figure 3 shows the result from a sample query into our image retrieval system. The query asked for images with 20% blue and 80% don't care and requested only the ten best matching images. Figure 3(a) shows the traditional way of displaying the resulting images as a one-dimensional list sorted by the distances from the query, while figure 3(b) shows the same images arranged according to a two-dimensional MDS. In the MDS display, similar images of desert scenes with yellowish ground group together at the top left, images with green plants group at the bottom, and the two other images – a desert image with a white ground and an image of a statue, are to the right. An all-blue image is comparatively dissimilar from the others, and is accordingly relegated to the far right. In this iterated-query framework, navigation can proceed by choosing a promising area in the MDS display and using a representative image out of that area as the next query.

4.2 Global MDS

Performing MDS on a large set of images can help the user understand the space of color images of the set. In figure 4 we see the MDS map of 500 images. It is easy to see that images group by their average chroma. For example, blue images are at the top-left, green images are at the top-middle, yellow images are at the top-right, and so forth. The images are also ordered from bottom-right to top-left by their average lightness, dark images are at the bottom-right and bright images are at the top-left. Higher dimensional MDS can be done on the image database where different characteristics of the images will be revealed, such as their average chroma (the projection of the images on the appropriate axes gives the chromaticity diagram), average lightness, the colorfulness of the images, and so forth. Now when we look for a sunset we see immediately where to go. At a glance, we can write off most of the database, and home in to the "sunset-looking" part of it. At the same time, we form a mental picture of the entire database. We see everything in coarse detail, and we have the impression of grasping the overall database content, at least in terms of color distributions. Given a joystick that lets us get closer to the area of interest, we have at the same time focus, because nearby images are large on the display, and context, because all or most other images are still visible at a distance. As we move about, we have the comforting impression that the whole database is there all the time, rather than being handed down to us in small fragments.

5 Conclusions

The methods presented in this paper open a novel set of tools and possibilities for image data-base navigation and visualization. The color signatures we have defined and the earth mover's distance between them seem to capture well the perceptual similarity or dissimilarity of images based on their color content. Furthermore, the low-dimensional geometric embeddings we compute using MDS techniques provide an intuitive way for the user to refine his/her query and to continue exploring interesting neighborhoods of the image space — or

to see large portions of it all at once.

All image query system are ultimately based on computational approximations to perceptual image distance — approximations whose quality we are often asked to take for granted. Our approach appears to be the first one to allow the user to explore, in an intuitive way, the area of the image space beyond what the system considers the neighborhood of the query. Such an exploration can provide increased confidence that what is wanted will not be missed.

Clearly much remains to be done. It is likely that distances between similar images provide much more information than distances between images which have little in common. Yet currently we compute large distances as accurately as small ones. As indicated in Section 3, we can gain significant speed-ups by simply using lower bounds for the earth mover's distance when the corresponding images are far apart. A major extension of our work will be to apply the concepts of signature and the earth mover's distance to other modalities which also convey information about the content of the image, such as shape and texture. The principle of our approach will remain that we measure the distance between images by the minimum 'work' needed to make their signatures the same. Thus the data in a signature need not be fully homogeneous, as long as we provide a set of modification operations, with associated costs, for each type of data present. We consider this ability to combine different kinds of feature sets and modalities (both in building the image database index and in computing the appropriate geometric embeddings) to be a unique advantage of our approach.

For the intuitive use of the geometric embeddings computed by MDS methods, it is crucial that the 'axes of variation' be perceptually clear to the user. This worked well for us in the case of color, in part because we started from data in a geometric color space whose axes have a familiar significance. Getting the same effect in the case of shape and texture seems more of a challenge. We intend to explore how to 'advise' MDS algorithms about what are desired coordinate axes to use. We also need to study more the relations between the axes chosen by MDS for related or overlapping image sets. Knowing the correspondence between these 'local charts' (in the sense of topology) of the image space can greatly help in providing a globally stable and consistent sense of navigation.

Acknowledgments: The authors wish to acknowledge helpful discussions with Scott Cohen and Jorge Stolfi.

References

[Bach *et al.*, 1996] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C. Shu. Virage image search engine: an open framework for image management. In *SPIE Conf. on Storage and Retrieval for Image and Video Databases IV*, 2670:6–87, 1996.

[Bentley, 1975] J. L. Bentley. Multidimensional binary search trees used for associative searching. *CACM*, 18:509–517, 1975.

[Guibas and Tomasi, 1996] L. J. Guibas and C. Tomasi. Image retrieval and robot vision research at Stanford. In *ARPA IUW*, pp. 101–108, 1996.

[Hafner *et al.*, 1995] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Trans. on PAMI*, 17(7):729–735, 1995.

[Kruskal, 1964] J. B. Kruskal. Multi-dimensional scaling by optimizing goodness-of-fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.

[Niblack *et al.*, 1993] W. Niblack, R. Barber, W. Equitz, M. D. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, G. Taubin, and Y. Heights. Querying images by content, using color, texture, and shape. In *SPIE Conf. on Storage and Retrieval for Image and Video Databases*, 908:73–187, 1993.

[Pentland *et al.*, 1996] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: content-based manipulation of image databases. *Int'l J. of Computer Vision*, 18(3):233–254, 1996.

[Stolfi, 1994] J. Stolfi. Personal communication, 1994.

[Stricker and Orengo, 1995] M. Stricker and M. Orengo. Similarity of color images. In *SPIE Conf. on Storage and Retrieval for Image and Video Databases III*, 2420:381–392, 1995.

[Swain and Ballard, 1991] M. J. Swain and D. H. Ballard. Color indexing. *Int'l J. of Computer Vision*, 7(1):11–32, 1991.

[Werman *et al.*, 1985] M. Werman, S. Peleg, and A. Rosenfeld. A distance metric for multi-dimensional histograms. *Computer Vision, Graphics, and Image Processing*, 32:328–336, 1985.

[Wyszecki and Styles, 1982] G. Wyszecki and W. S. Styles. *Color Science: Concepts and Methods, Quantitative Data and Formulae.* John Wiley and Sons, New York, NY, 1982.

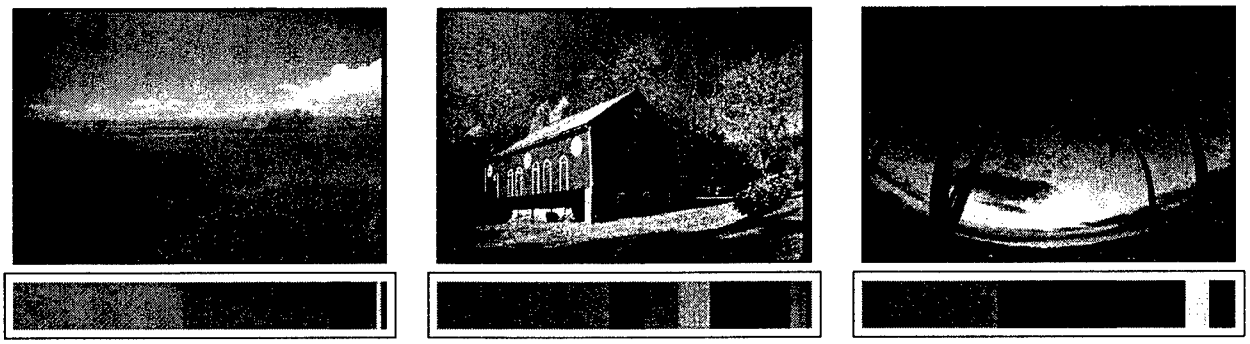
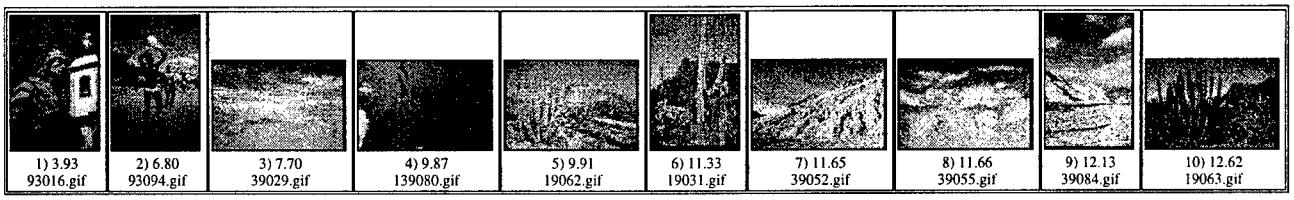
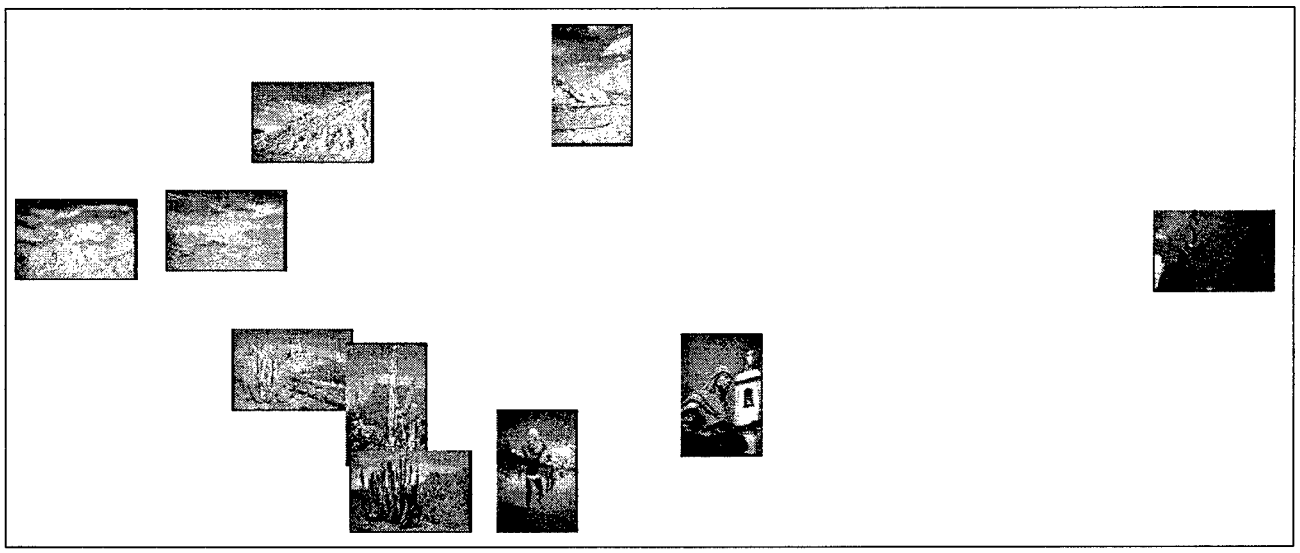


Figure 2: Three (color) images together with their color signatures. The left image contains mostly greens and blues, the middle image contains mostly greens, blues and browns, and the right image contains mostly yellows, browns and blacks. Color versions can be viewed at `http://vision.stanford.edu/irs/colorpics.html`.



(a)



(b)

Figure 3: The top ten images for a query that asked for 20% blue and 80% don't care. (a) Traditional display. (b) MDS map. Color versions can be viewed at `http://vision.stanford.edu/irs/colorpics.html`.

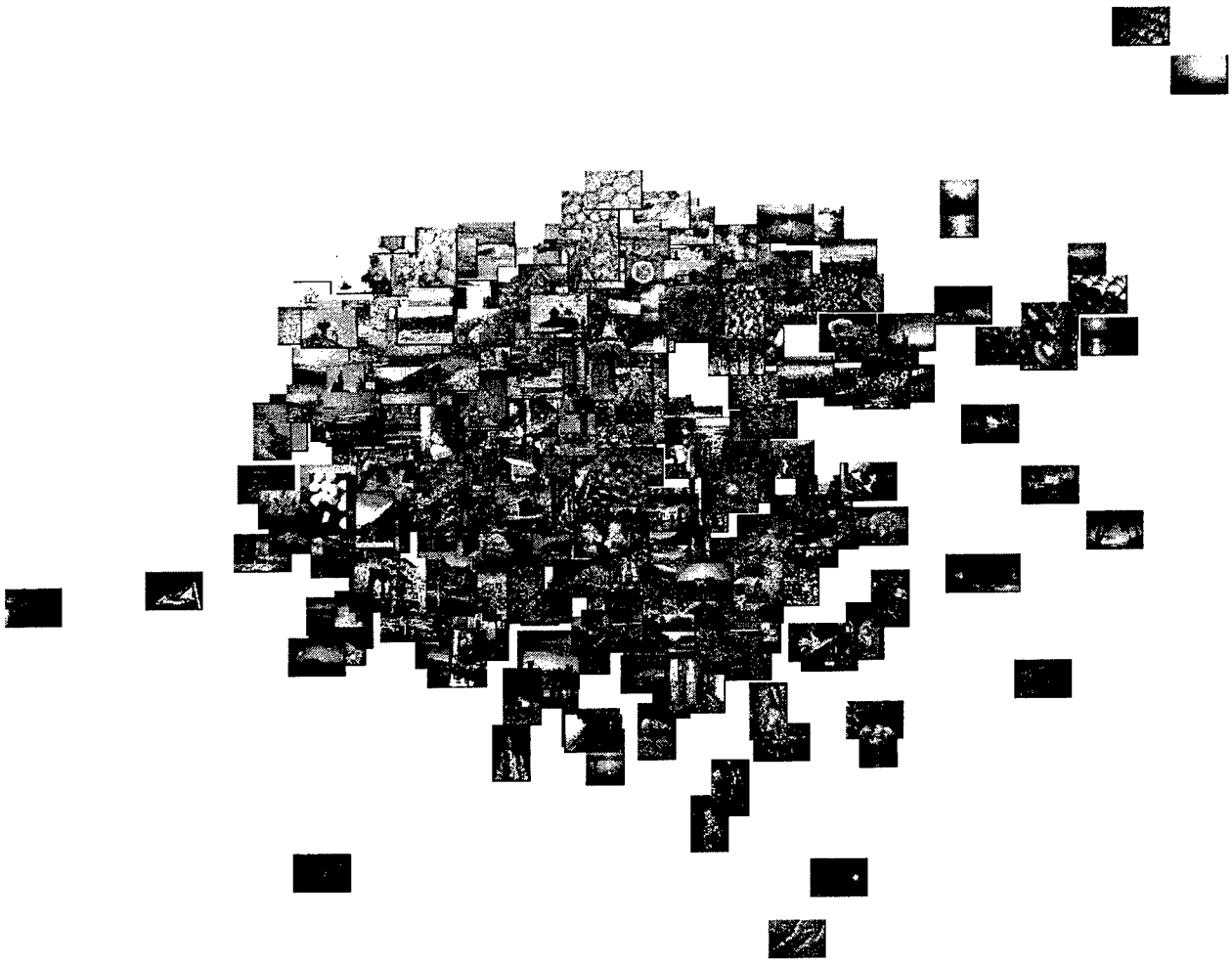


Figure 4: 2D MDS map of 500 images. A color version can be viewed at `http://vision.stanford.edu/irs/colorpics.html`.

Shape-based Image Retrieval Using Geometric Hashing*

Scott D. Cohen **Leonidas J. Guibas**
Computer Science Department, Stanford University
Stanford, CA 94305
[scohen,guibas]@cs.stanford.edu
http://robotics.stanford.edu/users/[scohen,guibas]/bio.html

Abstract

We present a general strategy for *shape-based image retrieval* which considers similarity modulo a given transformation group $\mathcal{G}$. The shape content of an image is summarized by recording what geometric primitives, such as line segments and circular arcs, fit where in the image. Geometric hashing is used to compute a set of primitive features which are invariant under a $\mathcal{G}$-transformation of the image. Our search engine is *feature-based* in the sense that similarity is determined by looping over the features in the query and asking: Which database images have features that are close to a given query feature? The most similar database images are ones that have many features which are close to query features. We apply our approach to an example database of 500 chinese character bitmaps.

1 Introduction

The function of a content-based image retrieval system [Niblack *et al.*, 1993, Guibas and Tomasi, 1996] is typically to find database images that look similar to a given query image or drawing. Database and query images are usually summarized by their color, shape, and texture content. Here we use the term *images* in a very broad sense that includes any type of graphical information. Examples of images include color or grayscale pixel images, technical drawings of aircraft parts, architectural drawings, line art, and figures produced with standard drawing programs.

In this paper, we focus on *shape-based image retrieval*. We consider the *shape content* of an image to be a set of planar curves that help identify the image. A set of curves which summarize an image will be called an *illustration* of that image. For example, we might perform edgel dectection and linking to obtain an illustration of a grayscale pixel image. Of course, the database image itself may already be an illustration. This is the case for an image which is a technical drawing. The shape index of an image is derived from an illustration of that image.

This paper presents a general strategy for shape-based image retrieval in which the similarity of images is considered with respect to some given transformation group. Accounting for transformations is necessary to handle illustrations which are produced from a variety of sources. Such illustrations are likely to be expressed in coordinate systems with different units for scale and position. If we want to retrieve a portrait image from its landscape version, then our notion of similarity must also account for differences in orientation. If the illustrations are extracted from imaged objects, then allowing for a projective transformation in judging similarity is important.

There are several different approaches to building a search engine. A straightforward approach is to compare every database image to the query using some dissimilarity function defined on pairs of images. Such a retrieval strategy will eventually become too slow for interactive use as the number of images in the database grows. A related approach is to summarize image content by a point/vector in $\mathbf{R}^d$ in such a way that the L_2 distance between points is a measure of image dissimilarity. A Euclidean-space nearest-neighbor algorithm may then be used to avoid brute force search. The problem with this approach is that the dimension needed

*This work was sponsored by the Defense Advanced Research Projects Agency under contracts DAAH04-94-G-0284 and DAAH01-95-C-R009, and by NSF grant CCR-9215219. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the United States Government, or Stanford University.

to capture differences in image content is likely to be quite high, perhaps even in the thousands, while current linear-space nearest-neighbor algorithms are limited in practice to maximum dimension $d_{\max} \approx 30$ due to "constant factors" which are exponential in the dimension. Yet another idea is to cluster database images so that when a query is far from a cluster representative, it will be far from all other images in the cluster. This allows the search to eliminate all the images in a cluster by comparing the query to only the cluster representative. This pruning strategy can be made precise with the triangle inequality when the dissimilarity function is a metric. Unfortunately, the pruning power of the triangle inequality decreases as the dimension increases. All the previously mentioned retrieval strategies are *image-based* in the sense that direct comparisons are made between images using a dissimilarity function.

Our retrieval strategy is *feature-based* in the sense that the similarity is determined by looping over the features in the query and asking: Which database images have features that are close to a given query feature? The most similar database images are ones that have many features which are close to query features. We have traded one nearest-neighbor problem in a high-dimensional *image space* for many nearest-neighbor problems in a low-dimensional *feature space*. The challenge is to find a small feature set of an image which captures the content the image. This problem is even more difficult when the features are required to be invariant under some transformation(s) of the underlying image.

Our feature extraction approach starts with an illustration of the image. The illustration curves are projected onto a basis of *basic shapes* such as line segments, corners, circular arcs, etc.. More precisely, we record *what basic shapes fit where* in the illustration curves. This strategy was first suggested in [Cohen and Guibas, 1996]. The projection step is discussed further in section 2. An invariant set of geometric primitives (a.k.a. basic shapes) is then derived from the *projected illustration* using *geometric hashing* ([Lamdan and Wolfson, 1988]). The invariance is with respect to a transformation of the projected illustration. The geometric hashing step is the subject of section 3. Define the *features* of an image to be the geometric primitives in its *invariant projected illustration*. The final preprocessing step is to build a nearest-neighbor search structure on the set of all features of all database images. Section 4 is devoted to the creation and use of the feature space nearest-neighbor structure. Finally, we conclude in section 5 with some problems that need to be addressed in future work.

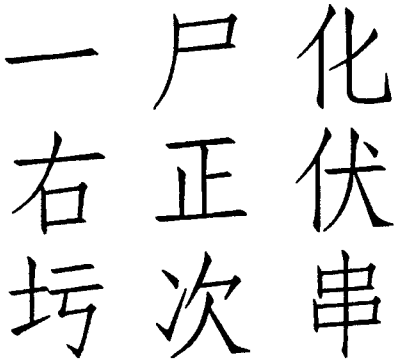


Figure 1: A small sample of images in a database of 500 chinese characters. The images are bitmaps.

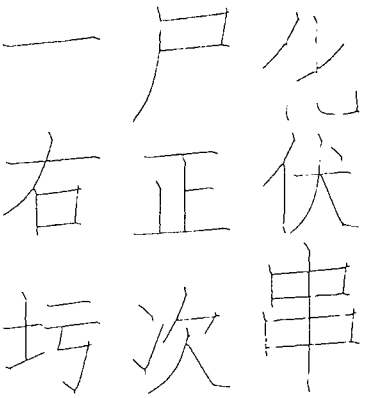


Figure 2: The shape summary of a chinese character bitmap is the medial axis of the set of black pixels which define the character.

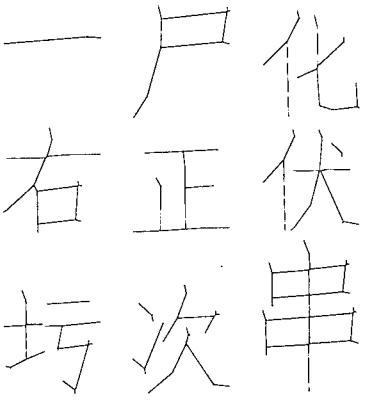


Figure 3: The chinese character illustrations are projected onto a basis with a line segment as the only basic shape.

The strategy outlined in this paper will be applied to an example database of 500 chinese characters. A small sample of images in this database is shown in figure 1.

The collection of chinese characters is an ideal database to test our ideas because there are many patterns which occur throughout the database at different scales, locations, and orientations. Our shape summary of a character is the *medial axis* of the set of black pixels which define the character. The results shown in figure 2 were computed using algorithms and software described in [Ogniewicz and Kübler, 1995]. The character skeletons are a very good one-dimensional summary of the characters.

2 Projecting an Illustration onto a Basis of Basic Shapes

If an illustration is not created by a drawing program with a palette of geometric primitives, then its curves are likely to be polylines with a large number of vertices. This is the resprresentation for the medial axes shown in figure 2, as well as any illustration produced by linking edgels. Higher level descriptions of such curves will greatly simplify the indexing process. Therefore, we project the illustration onto a basis of basic shapes such as line segments and circular arcs. The projected illustration is a union of basic shapes which approximate the illustration curves. The basis of basic shapes is chosen so that as little information as possible is lost during projection. Different databases may call for different bases.

There are many methods for finding common geometric primitives in polylines. For example, the segmentation algorithm in [Lowe, 1987] uses a split-and-merge algorithm to divide an edgel chain into straight segments. The FEX algorithm in [Etemadi, 1992] finds straight segments and circular arcs, while the algorithm in [Rosin and West, 1995] identifies straight segments, circular arcs, and elliptical arcs. The algorithm in [Cohen and Guibas, 1997] locates any pattern shape described as a polyline within another polyline, allowing for a similarity transformation of the pattern.

There is a potential problem with separating the curve extraction and curve projection steps. The algorithms mentioned above operate on one polyline at a time with no regard for the union of polyline curves as a whole. If a long straight line segment is part of two different polylines in the illustration, then it will not be found. In the case when the underlying image is a color or grayscale pixel image, one could use an algorithm for finding geometric primitives that works directly on the pixel data.

The chinese character illustrations are well approximated using circular arcs and line segments, but we simplify the medial axis pixel chains into line segments only. The results are shown in figure 3. A naive polyline simplification algorithm was used to approximate the medial axis pixel chains by straight segments. Consider the error in approximating the polygonal chain between start vertex u and end vertex w by the line segment $\overline{uw}$ connecting u and w. If this error is within a given bound, then reset w to the vertex right after w in the chain, and try the next segment. If the error exceeds the given tolerance, then approximate the chain from u to the vertex v just before w by the line segment $\overline{uv}$, reset the start vertex u to v, and try to find a line segment approximation starting from the new u. A reasonably high error bound was used in order to segment the medial axis chains into a small number line segments.

3 Accounting for Transformations Using Geometric Hashing

As mentioned in the introduction, similarity of images is considered with respect to some given transformation group $\mathcal{G}$. Ideally, a transformation of the underlying image will cause the same transformation of the corresponding illustration and projected illustration. We cannot directly compare two projected illustrations to judge image similarity. Instead, we derive an invariant feature set from the projected illustration using geometric hashing. This technique will produce the same feature set given projected illustrations S and $g(S)$, where $g \in \mathcal{G}$.

Geometric hashing is a method used to compare two point sets under some transformation group. Usually, the method is applied to finite point sets $P = \{ p_1, \ldots, p_m \}$ and $Q = \{ q_1, \ldots, q_n \}$. We illustrate the basic idea with the case of comparing P and Q under the group of translations. Consider the sets

$$\begin{aligned} I_i(P) &= \{ p_k - p_i \,:\, 1 \le k \le m,\ k \ne i \} \\ I_j(Q) &= \{ q_l - q_j \,:\, 1 \le l \le n,\ l \ne j \}. \end{aligned}$$

Note that $I_i(P)$ and $I_j(Q)$ are invariant under translation of P and Q, respectively. If translating the set P by $q_j - p_i$ produces a good match between P and Q, then the two sets $I_i(P)$ and $I_j(Q)$ will match well. The method can be made robust to missing data by comparing the trans-

lation invariant sets

$$I(P) = \bigcup_{i=1}^{m} I_i(P) \qquad \text{and} \qquad I(Q) = \bigcup_{j=1}^{n} I_j(Q).$$

In words, each of the points of P is recorded in $m-1$ different coordinate systems. The ith coordinate system has the same orientation and scale as the coordinate system of P, but its origin is at the point p_i. To compare P to Q, we compare $I(P)$ to $I(Q)$. Note that the sizes $|I(P)| = m(m-1) = O(m^2)$ and $|I(Q)| = n(n-1) = O(n^2)$ are quadratic in the original set sizes.

The details for the translation case can be generalized to other transformation groups. The general idea is to use subsets of P as bases in which to record all the other points in P. Ordered pairs of points (p_i, p_j) define the basis in the case of similarity transformations. The segment $p_i\vec{p}_j$ plays the role of the unit interval e_1 from $(0,0)$ to $(1,0)$ in recording the other points of P with respect to (p_i, p_j). More precisely, let T_{ij} be the transformation which maps $p_i\vec{p}_j$ to e_1. Then recording the point p_k with respect to (p_i, p_j) means recording $T_{ij}(p_k)$. The total number of points in the invariant set $I(P)$ is $O(m^3)$ in this case. For affine transformations, an ordered triple (p_i, p_j, p_k) defines the basis in which to record the other points in P. If T_{ijk} is the transformation that maps (p_i, p_j, p_k) to the vertices $(0,0)$, $(0,1)$, and $(1,0)$ of the right triangle Δ_1, then recording p_l with respect to (p_i, p_j, p_k) means recording $T_{ijk}(p_l)$. The total number of points in the invariant set $I(P)$ is $O(m^4)$ in this case.

The projected illustrations for our chinese character database are sets of segments. Although we do not have finite point sets, we can still apply the idea of geometric hashing to obtain a feature set which is invariant to a similarity transformation of the projected illustration. This is done by allowing each segment in the projected illustration P to play the role of the unit interval e_1. If P contains m segments, then we will have $2m$ different coordinate systems in which to record the segments in P (the factor of two is from considering both orderings of the segment endpoints). Therefore, using segment endpoints as basis point pairs leads to an invariant set $I(P)$ of $O(m^2)$ segments. The set $I(P)$ consists of m copies of P at varying scales, locations, and orientations. The overlap that occurs among these copies makes it very difficult to see the individual copies. A picture of some $I(P)$ from the chinese character database is not very informative, and hence no figure is provided.

4 Searching the Database

The final preprocessing step is to build a nearest-neighbor search structure on the feature space. When a query is given, its features are extracted in exactly the same manner as for the database images. For each query feature, we query the nearest-neighbor search structure for the k nearest database features to the query feature. Each time a database image has a feature which is close to a query feature, its similarity score is increased. As the similarity scores are updated, the R greatest image scores (and corresponding images) are tracked. Once all the query features have been processed, the R images with the highest similarity scores are returned.

An ideal situation for finding nearest database features is when the database features are points in a low-dimensional space, and the the L_2 distance between points measures feature dissimilarity. In this case, a standard Euclidean-space nearest-neighbor search strategy may by employed. The features in the chinese character database are the line segments in the invariant projected illustration. What is an appropriate distance measure between two line segments? We might, for example, use the Hausdorff distance between two line segments. There is work [Yianilos, 1993, Brin, 1995] on nearest-neighbor searching in general metric spaces (i.e. using only the distance between two objects). Here we opt for a simpler, ad hoc approach which embeds the segments as points in a four-dimensional Euclidean space. A directed line segment is specified by a quadruple

$$(l, \theta, a, b),$$

where l is length of the segment, θ is the angle the segment makes with the horizontal, and (a, b) is the position of the first endpoint. Note that the units of the components are different, so it does not make sense to use the L_2 distance unless we first normalize the components. Toward this end, we compute the standard deviations σ_l, σ_θ, σ_a, σ_b of the four component values over a large sample of database features. We use the L_2 distance between the normalized point features

$$\left(\frac{l}{\sigma_l}, \frac{\theta}{\sigma_\theta}, \frac{a}{\sigma_a}, \frac{b}{\sigma_b}\right)$$

as a measure of feature dissimilarity.

Our choice for a Euclidean nearest-neighbor searching algorithm is due to Arya, Mount, et.al. in [Arya *et al.*, 1994]. The algorithm preprocesses a set $S \subset \mathbf{R}^d$ of n points in $O(n \log n)$ time and $O(n)$ space, so that the k nearest

neighbors to a given query point q can be computed in $O(k \log n)$ time. We apply the algorithm to find k nearest features to a given query feature, where $k = 32$ (in this setting, $d = 4$). If we let F denote the total number of database features and f_Q is the number of features in the query Q, then our query time is $O(f_Q \log F)$. For our 500 image database with the features extracted as previously described, a typical query takes roughly one second on an SGI Indy. Some sample queries and results are shown in figure 4.

5 Some Problems for Future Work

Our feature-based algorithm uses a very *one-way* notion of distance. A query and database image are similar whenever the database image has many of the same features as the query. There is no penalty for extra information in a database image which might cause it to look quite a bit different from the query. A possible solution to this problem involves tagging each feature point with both the image and basis points that produced it. This will allow us to estimate the transformation which makes the query match a particular database image, as well as the fraction of unmatched arclength in the database image.

Unfortunately, there is a more serious problem with our overall approach. The similarity score depends heavily on the segment decompositions of the projected illustrations – and these decompositions are not canonical. A high similarity score will be obtained iff there is a similarity transformation of Q that makes many segments in Q match well many segments in P, where two segments match well iff both pairs of endpoints are close. This fact is due to our use of pairs of segment endpoints as bases in the geometric hashing step. In essence, we are judging the similarity of the representations of the projected illustrations instead of the projected illustrations themselves. This *representation problem* will be the subject of future research.

A third problem is that our geometric hashing strategy produces too many features to index. If there are m segments in a projected illustration, the invariant projected illustration will have $O(m^2)$ segments. This brute force approach is motivated by the fact that we want to be able to match a subset of the query to a subset of a database illustration without making any *a priori* assumptions about features that are likely to appear in both the database illustration and a similar query illustration. Suppose, instead, that we record segments only with respect to the c longest segments in the set, where c is a small constant. This strategy produces an invariant projected illustration with only $O(m)$ segments, but it assumes that a long segment in a database image is likely to appear as a long segment in a similar query. Thus, the representation problem remains.

Acknowledgements

We would like to thank Guillermo Sapiro for recommending the medial axis code package based on [Ogniewicz and Kübler, 1995], and Carlo Tomasi for carefully reading the manuscript and providing useful comments.

References

[Arya *et al.*, 1994] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu. An optimal algorithm for approximate nearest neighbor searching. In *Proc. of the Fifth Annual ACM-SIAM Symp. on Discrete Algorithms*, pp. 573–582, 1994.

[Brin, 1995] S. Brin. Near neighbor search in large metric spaces. In *Proc. of VLDB '95. 21st Int'l Conf. on Very Large Data Bases*, pp. 574–584, 1995.

[Cohen and Guibas, 1996] Scott D. Cohen and Leonidas J. Guibas. Shape-based illustration indexing and retrieval - some first steps. In *ARPA IUW*, pp. 1209–1212, 1996.

[Cohen and Guibas, 1997] Scott D. Cohen and Leonidas J. Guibas. Partial matching of planar polylines under similarity transformations. In *Proc. of the Eighth Annual ACM-SIAM Symp. on Discrete Algorithms*, pp. 777–786, 1997.

[Etemadi, 1992] A. Etemadi. Robust segmentation of edge data. In *Int'l Conf. on Image Processing and its Applications*, pp. 311–314, 1992.

[Guibas and Tomasi, 1996] Leonidas J. Guibas and Carlo Tomasi. Image retrieval and robot vision research at Stanford. In *ARPA IUW*, pp. 101–108, 1996.

[Lamdan and Wolfson, 1988] Yehezkel Lamdan and Haim J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Second Int'l Conf. on Computer Vision*, pp. 238–249, 1988.

[Lowe, 1987] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

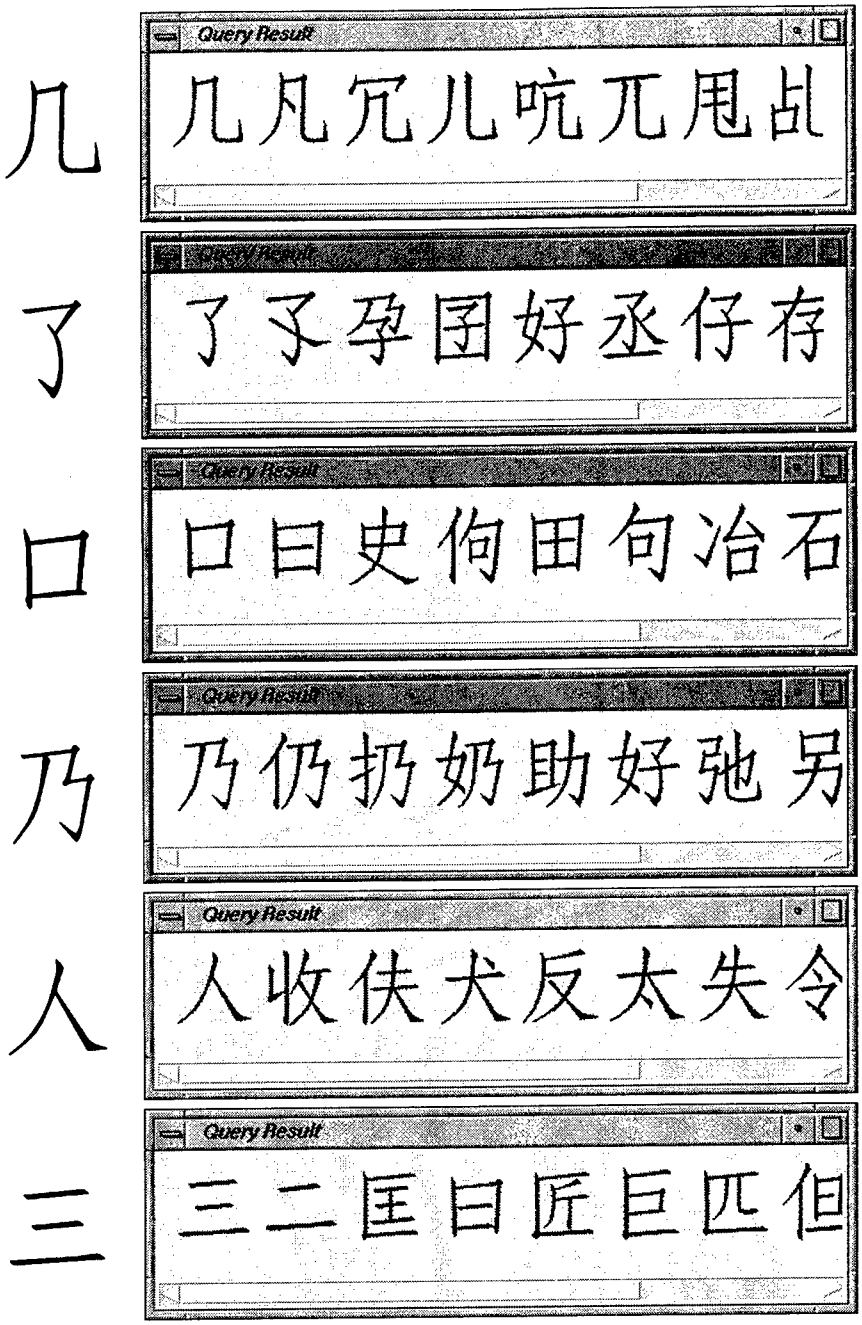


Figure 4: Sample queries (left) into the chinese character database, with corresponding results (right). Each query takes about one second on an SGI Indy.

[Niblack *et al.*, 1993] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: querying images by content using color, texture, and shape. In *Proc. of the SPIE*, 1908:173–187, 1993.

[Ogniewicz and Kübler, 1995] R. L. Ogniewicz and O. Kübler. Hierarchic Voronoi skeletons. *Pattern Recognition*, 28(3):343–359, 1995.

[Rosin and West, 1995] Paul L. Rosin and Geoff A.W. West. Nonparametric segmentation of curves into various representations. *IEEE Trans. on PAMI*, 17(12):1140–1153, 1995.

[Yianilos, 1993] P. Yianilos. Data structures and algorithms for nearest neighbor searching in general metric spaces. In *Proc. of the Fourth Annual ACM-SIAM Symp. on Discrete Algorithms*, pp. 311–321, 1993.

Configuration Based Scene Classification and Image Indexing

Pamela R. Lipson, Eric Grimson, and Pawan Sinha*
MIT Artificial Intelligence Lab, 545 Technology Square, Cambridge, MA 02139
E-MAIL: lipson@ai.mit.edu,welg@ai.mit.edu,sinha@ai.mit.edu

Abstract

Scene classification is a major open challenge in machine vision. Most solutions proposed so far such as those based on color histograms and local texture statistics cannot capture a scene's global configuration, which is critical in perceptual judgments of scene similarity. We present a novel approach, "configural recognition", for encoding scene class structure. The approach's main feature is its use of qualitative spatial and photometric relationships within and across regions in low resolution images. The emphasis on qualitative measures leads to enhanced generalization abilities and the use of low-resolution images renders the scheme computationally efficient. We present results on a large database of natural scenes. We also describe how qualitative scene concepts may be learned from examples.

1 The Problem

The goal of our work is to classify scenes based on their content. Scene classification has applications for the problem of image and video database indexing. With the increase in the number and sizes of digital libraries there is a need for automated, flexible, and reliable image search algorithms.

Several strategies have recently been proposed for image classification. Most use aggregate measures of an image's color and texture as a signature for an image that can be used to determine how similar one image is to another. Image database indexing systems based on this idea include QBIC [Ashley *et al.*, 1995] and VIRAGE[Bach *et al.*, 1996]. These similarity measures are adequate if the goal is to find images with similar distributions of color or other low level signal characteristics. However, if the goal is to find images from a given object/scene class, such as snowy mountains or waterfalls, the previously defined similarity measures often produce results incongruent with human expectations (Figure 1).

Figure 2 shows three images that perceptually belong to the same class, viz. coasts. However, the elements of which they are composed vary significantly in color distribution, texture, illumination, and spatial layout.

*This work was sponsored in part by ARPA under ONR contract N00014-95-1-0600

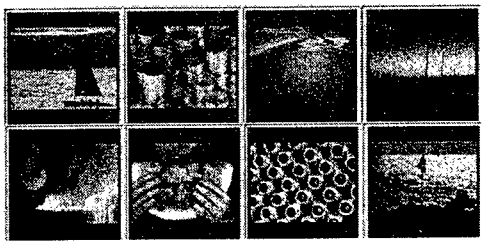


Figure 1: Using color histograms to find the most similar images to a water scene at sunset (upper left) returns pictures of money, molten liquid, and a woman eating watermelon. Although these images all have the same overall golden color, most differ greatly in semantic content.

Figure 2: These images all belong to the coastal images class although colors, illumination, and layout vary widely.

In this paper, we suggest a novel representational strategy, "configural recognition", as a partial solution to the scene classification problem. The strategy encodes class models as sets of qualitative relationships between low resolution image regions. We will demonstrate how models of this form can tolerate within class variations while also discriminating between classes.

2 Motivation for the approach

Our approach to scene classification is motivated by three considerations derived from studies of human perception.

1) The importance of global scene configuration. In Figure 3 the image on the right has been derived from the one on the left by dividing the latter into pieces and permuting their positions. Clearly, both images have identical chromatic and (to a large extent) textural statistics. Yet, perceptually, they do not belong to the same class since they have different overall configurations. This observation has been replicated in several systematic psychological studies which demonstrate that a stimulus in correct spatial configuration allows for more accurate and rapid detection of itself or its parts than the

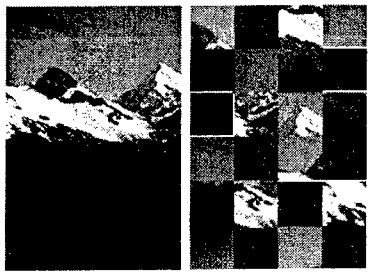


Figure 3: A mountain picture and its scrambled counterpart. Although both images contain the same color and textural characteristics, perceptually we would not classify the second image in the same category as the first.

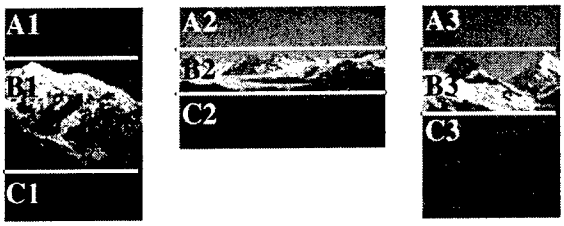


Figure 4: Three snow-capped mountains are shown. Each is divided into three regions (A, B, C). Perceptually, the corresponding regions have similar content, even though they differ in their absolute sizes, positions, and colors.

same stimulus with incorrect spatial relationships [Bar and Ullman, 996][Biederman, 1972]. Our conclusion is that the overall organization of a scene's parts strongly influences its interpretation.

2) The use of qualitative measurements. Figure 4 shows three snow-capped mountain scenes. This class of images may be described as having three perceptually salient regions: a blue region (A), a white region (B), and a grey region (C). In all cases region A is above region B which is above region C. Therefore, even though the particular instances of the class exhibit these regions at diverse absolute locations and over different spatial extents, one constant is that all the regions across the images have the same *relative* spatial layout.

Just as relative spatial relationships may be used to encode the overall configuration of scene content, relative photometric relationships between image regions may also be important for perceptual classification of scenes. For instance in all the images in figure 4, the regions labeled A are consistently bluer and brighter than those labeled C.

This suggests that the classification of a scene may remain valid as long as the relative relationships between the image regions remain the same, even though the absolute region values may change. However, when the ordinal relationships are violated, often the percept and therefore the classification of that image is greatly altered. The difficulty observers experience in recognizing photographic negatives is a case in point.

Figure 5: Low resolution images may be sufficient for recognition. These images are identifiable despite their extremely poor resolutions.

3) The sufficiency of low spatial frequency information for scene classification. Figure 5 shows several readily recognizable low resolution thumbnails. The only information retained in these small images is an arrangement of low frequency photometric regions. This observation suggests that we can base our classification algorithm on an image's low frequency information.

3 Qualitative encoding of scene structure

The configural recognition scheme encodes class models as a set of salient low frequency image regions and salient qualitative relationships between those regions. The most closely related work to what we are about to describe is the ratio-template construct devised by Sinha to detect faces under varying illumination conditions. The construct consists of relative luminance relationships between image regions with fixed spatial positions [Sinha, 1994]. Some researchers have previously considered using qualitative spatial relationships in the context of scene classification to describe the relationships between objects or object subparts in images [Chang and Lee, 1991][Petrakis and Faloutsos, 1994]. They have typically assumed, however, that the objects or object subparts are labeled by hand or easily identifiable.

The configural recognition system differs from these approaches by constructing class models for scenes from a wide vocabulary of relative relationships, including both spatial and photometric, between image regions. In the current system, class models are described using seven types of relative relationships between image patches. Each of these relationships can have the following values: less than, greater than, or equal to. The first three relations encode the relative color between image regions in terms of their red, green, and blue components. The fourth relationship used is relative luminance between the patches. The spatial relationships used are relative horizontal and vertical descriptions with respect to the upper left corner of the image and the cardinal

axes. We also encode relative size, where the size of the patch is described by how many pixels it covers.

Figure 6(a) denotes example beach scenes. (The example beach scene shown when rendered in color has blue sky, green water, and tannish colored sand). Figure 6(b) shows three highlighted image patches, from an image grid of large equally sized patches, and their relative relationships, denoted by arrows. This constitutes one possible model for beach scenes. The relationships in the model are that there is a bluer region, which is above and to the right of a greener region, both of which are above a more tan and lighter region.

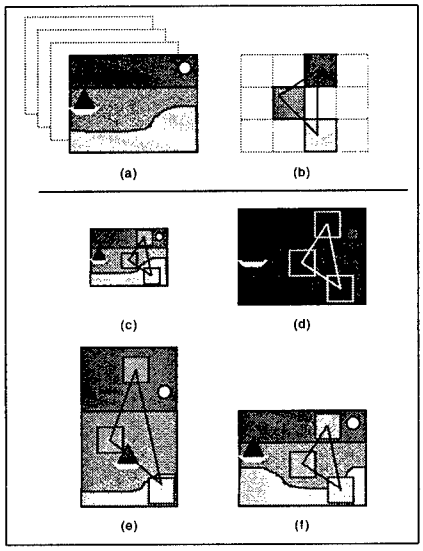


Figure 6: (a) A representative set of example beach images. (The example beach scene when rendered in color has blue sky, green water and tannish colored sand.) (b) One encoding of a qualitative beach concept. This model remains valid over many commonplace scene variations including (c) scale changes, (d) illumination variations (the colors have changed but the color and spatial relationships between the patches remain the same), (e) differing viewing parameters (distal vs. close up view), and (f) geometry changes.

3.1 Benefits of qualitative encoding

There are at least four significant benefits to using low-frequency image regions and their relative relations to encode scene classes. 1. *Invariance to many scene transformations.* The prime benefit is that the use of relative relationships over low frequency patches allows the system to describe class similarities even though the exemplars may differ in appearance due to various lighting conditions, viewing positions, and other scene parameters. Figure 6(c-f) illustrate how the relative relationships encoded in the model (shown in figure 6(b)) remain valid over different but very commonplace image distortions such as changes in scale, illumination, viewing parameters and geometry. 2. *Immunity to high-frequency sensor noise.* 3. *Dimensionality reduction.* Instead of having to use high-resolution images, 32x32 thumbnails sufficed for the classification task. 4. *Simple image partitioning requirements.* Partitioning the image with a uniform grid suffices.

4 Model to image matching

We can think of the model as a prototype of a class. When the model is compared to the image, the model can be deformed by moving the patches around so that the model best matches the image in terms of relative luminance and photometric attributes without violating the encoded relative spatial arrangements. The model in this sense acts as a deformable template. A match between the model and a subset of the image can be defined by how well the deformed model matches the image subset and how little deformation was required to find that match.

5 Implementation and testing

The configural approach to scene classification was tested by generating several class templates and subsequently using these models to classify a large database of natural images. For each template, the automated classification was reported as a binary decision of either a member or non-member of the class. We compared the results of the template classification to perceptual class judgments made by human observers.

The test database consists of 700 images from prepackaged CD-ROM collections from Corel which contained 100 images each with titles such as "Fields" and "Glaciers and Mountains". The total collection contains pictures which have a wide range of content, colors, textures, viewing positions, and weather conditions. Although the images in these compilations were mostly of natural scenes, many contain people, animals, and man-made structures such as fences, houses, and boats.

Each image in the database was iteratively smoothed and subsampled to create a three tier Gaussian pyramid of low resolution images of sizes 32x32, 16x16, and 8x8 pixels.

We manually constructed class templates for snowy mountains, snowy mountains with lakes, fields, and waterfalls. We compared each template to the datatbase of low resolution images. In figure 9 we describe the snowy mountain template and the waterfall template and show pictorially the database retrieval results using these class models (figures 10- 15). In table 1 we describe the results of all four templates on the database in terms of "true positives", "false positives". (See [Lipson, 1996] for more details.)

6 Learning the scene concept

We have demonstrated that models consisting of qualitative relationships between low frequency image regions can be used effectively to classify images.

Table 1: The classification results from four hand crafted templates.The results are reported in terms of the "true positives" and "false positives" with respect to human perceptual classification of the 700 image database.

RESULTS	"true pos."	"false pos."
Snowy mount.	75%	12%
S. mnt. w/lake	67%	1%
Field	80%	7%
Waterfall	33%	2%

It would be desirable if instead of hand-crafting the models, an automated process could take a set of example images and generate a set of templates which describe the relevant consistencies between the pictures in the example set.

We have developed an algorithm that computes the consistent relationships between regions across a set of example images. The algorithm first computes all pairwise qualitative relationships between each low resolution image region. For each region, the algorithm also computes a rough estimate of its color from a coarsely quantized color space as a measure of perceptual color. The next step, for each region, groups the image into directional equivalence classes, such as "above" and "below", with respect to that region. Redundant relationships for each region in each equivalence class are eliminated. The next step is to compute the consistent set of region relationships across the set of examples. There is, however, a problem in that the correspondence of regions across the images is not known. A reasonable assumption is that corresponding regions in each image are likely to occur in similar positions. To determine the set of consistent relationships, we only compare the set of relationships/colors in a neighborhood surrounding each region location across all the example images.

We tested the approach by generating randomly colored synthetic images. A three patch qualitative concept was embedded in each image. The absolute colors and positions of the patches in the concept were allowed to vary as long as the qualitative color and spatial relationships were not violated. Figures 7 and 8 respectively show the inputs to and output from the learning system. The extracted concept *matched* the original randomly generated concept. We are currently testing this approach on real imagery. We are also extending our algorithm to allow the user to delineate particularly salient regions in the images.

7 Summary and Conclusions

We have presented a novel approach to classifying scenes in terms of qualitative relationships between low frequency image regions that provides a computationally efficient way to encode overall scene structure. Although the configural recognition approach

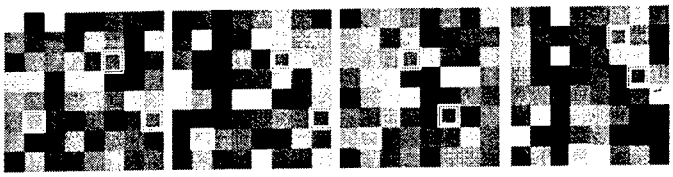


Figure 7: Four example input images to the learning algorithm. The patches which correspond to the embedded qualitative concept are highlighted in white in each image.

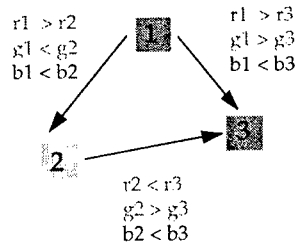


Figure 8: Resulting qualitative concept determined by the learning algorithm. The learned concept matches the concept embedded in each of the images.

appears to be a promising strategy for scene classification, it also has limitations. For instance, the technique is not suited to make fine quantitative discriminations, describe classes of functionally defined objects, nor to classify scenes which depend on specific object recognition. We are experimenting with an expanded repertoire of qualitative and quantitative information for classification of a broader class of images. For instance, we have created a template which includes relative texture measurements in order to classify cityscapes (see figure 16).

References

[Ashley *et al.*, 1995] J. Ashley, M. Flickner, D. Lee, W. Niblack, and D. Petkovic. Query by image content and its applications. IBM Research Report, RJ 9947 (87906), Computer Science/Mathematics, March, 1995.

[Bar and Ullman, 996] M. Bar and S. Ullman. Spatial context in recognition. *Perception*, vol. 25, pp. 342-352, 1996.

[Bach *et al.*, 1996] J.R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R.C. Jain, and C. Shu. Virage image search engine: an open framework for image management. *SPIE Storage and Retrieval of Image and Video Databases*, Vol. 4, pp. 76-87, 1996.

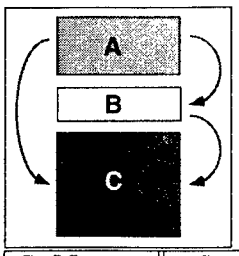
[Biederman, 1972] I. Biederman. Perceiving real world scenes. *Science*, Vol. 177, pp. 77-80, 1972.

[Chang and Lee, 1991] C. Chang, S. Lee, Retrieval of similar pictures on pictorial databases. *Pattern Recognition*, Vol. 23, No. 7, pp. 675-680, 1991.

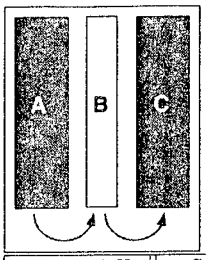
[Lipson, 1996] P. Lipson. Context and Configuration Based Scene Classification. *Ph.D. Thesis*, MIT, Sept. 1996.

[Petrakis and Faloutsos, 1994] E. Petrakis and C. Faloutsos. Similarity searching in large image databases. CS Tech. Report 3388, U. Maryland, College Park, Dec. 1994.

[Sinha, 1994] P. Sinha, Image invariants for object recognition. *Invest. Opth. and Vis. Science*, 34/6, 1994.



S. Mount.	Spat.	Lum.	Color
A to B	$A_y < B_y$	$A_l < B_l$	$A_r < B_r$
			$A_g < B_g$
			$A_b < B_b$
A to C	$A_y < C_y$	$A_l > 1.2C_l$	$A_b > C_b$
B to C	$B_y < C_y$	$B_l > C_l$	$B_r > C_r$
			$B_g > C_g$
			$B_b > C_b$
A to A			$A_b > A_r$
			$A_b > A_g$



Waterfall	Spat.	Lum.	Color
A to B	$A_x < B_x$	$1.6A_l < B_l$	$A_r < B_r$
			$A_g < B_g$
			$A_b < B_b$
B to B		$B_l > 80$	$B_r < 1.2B_b$
			$B_r < 1.2B_g$
B to C	$B_x < C_x$	$B_l > C_l$	$B_r > C_r$
			$B_g > C_g$
			$B_b > C_b$

Figure 9: Qualitative models for two natural-scene classes, viz. snowy mountain scenes and waterfalls. The figures on top show the schematic layouts of the models while the tables list the qualitative constraints that inter- and intra-region relationships in a given image have to satisfy for the image to be accepted as a member of the scene class.

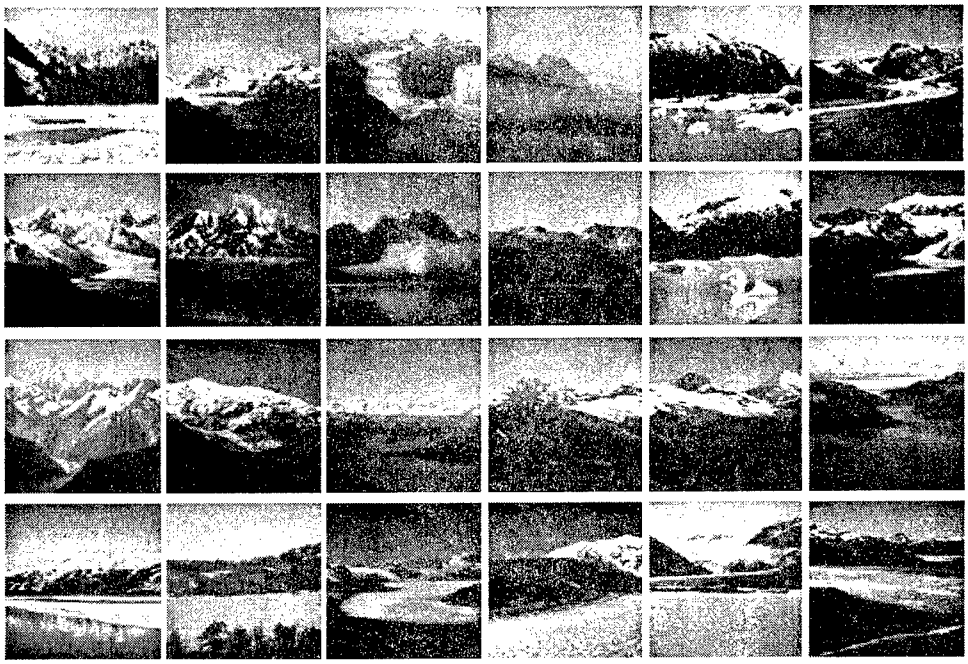


Figure 10: "True positives" detected by the snowy mountain template. Notice the diversity in these scenes captured by a single qualitative model.

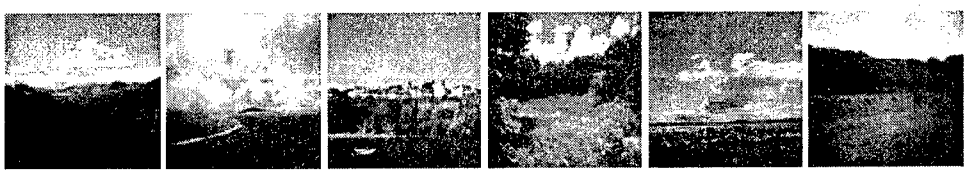


Figure 11: "False positives" detected by the snowy mountain template. Since the qualitative model does not encode fine textural details, it sometimes fails to distinguish between snowy mountains and white clouds.

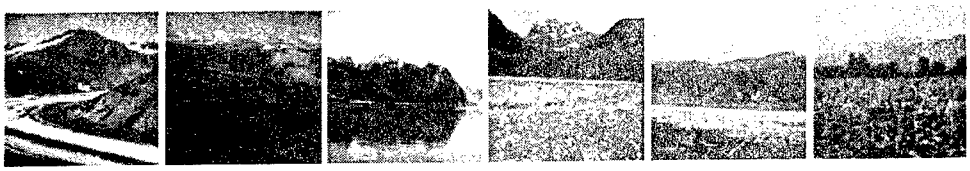


Figure 12: Mountain scenes not detected by the snowy mountain template. These failures are often due to significant differences between image configurations and the general scene structure encoded in the model. Also, sometimes the scene entities such as the snowy mountains are too small to be picked up by the qualitative model in the low frequency images.

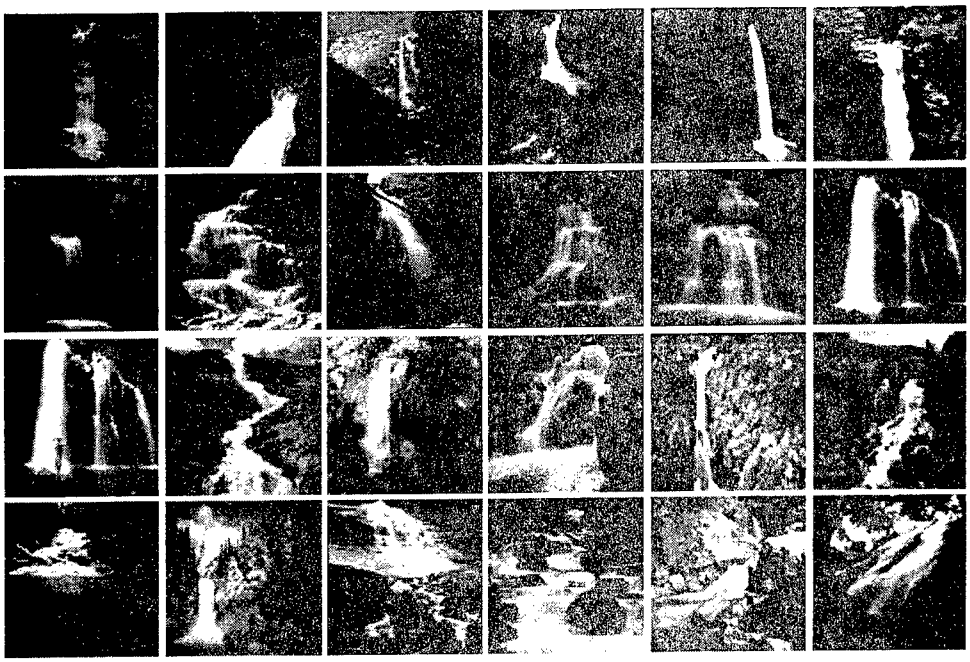


Figure 13: "True positives" detected by the waterfall template.

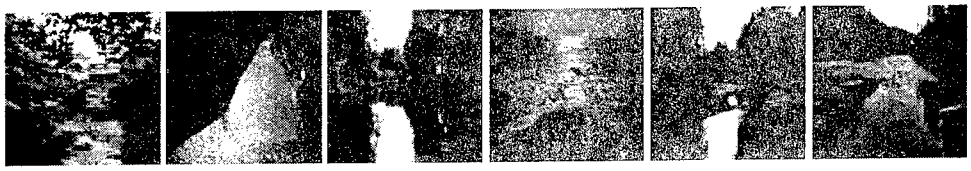


Figure 14: "False positives" detected by the waterfall template.

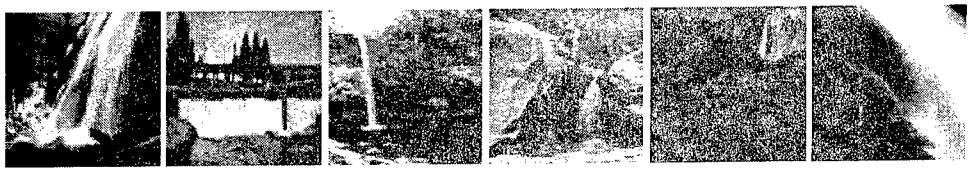


Figure 15: Waterfall scenes not detected by the waterfall template.

Figure 16: A demonstration of the use of relative textural statistics in the configural recognition framework. These cityscape scenes were detected by a qualitative mode that encoded not only qualitative chromatic and spatial relationships but also ordinal relations between the orientation energies in different image regions.

Extracting Templates for Scene Classification using a Few Examples

A. Lakshmi Ratan and W.E.L. Grimson*
Artificial Intelligence Lab, MIT
545 Tech. Square, Cambridge, MA 02139
E-MAIL: aparna@ai.mit.edu
HOMEPAGE: http://www.ai.mit.edu/people/aparna

Abstract

We present a method for classifying images in a databases where the query is a small set of images that represent the class. The algorithm extracts the dominant relative color, luminance and spatial relations between image patches in the low resolution query images and uses these relations to build one or more flexible templates. The templates capture the relative spatial and color properties of the class and are matched against the database to retrieve images that belong to the same class. Our experiments show that the algorithm that builds these templates automatically from a set of examples is fast, requires little storage and reliably classifies images of natural scenes. These templates can be further refined to obtain more selectivity by using an interactive process where the user picks the desired images from the general set returned at the first stage and the system repeats the process at a higher resolution.

1 Introduction

We investigate a method for learning relational templates that capture the luminance,color and spatial properties of a class of natural scene images given a set of examples that belong to the class. The templates that are extracted contain information about the relative color and luminance properties and the spatial layout of the class of images described by the example set. They represent a user-defined query-class. The templates extracted from the query images can then be matched against the entire database to obtain images that belong to the same class. This system allows the user to define the class using example images rather than hand-drawn representations or other abstract queries and also allows the user to refine the query and perform a more selective search based on the initial set of matches that it returns.

1.1 Image Classification Systems

In the past few years, the large number of digital image and video libraries has led to the need for flexible, automated content-based image retrieval systems which can efficiently retrieve images from a database that are similar to the user's query. Since what a user wants can vary greatly, we also want to provide a way for the user to explore and refine the query by letting the system bring up examples at every stage.

Typical techniques that have been proposed to search through these large image databases efficiently use simple image properties like color histograms and color layout([Swain *et al.*1991] [QBIC 1995]), compact representations of the image texture properties[Virage 1996], [Pentland *et al.*1996], spatial information and information encoded by the dominant wavelet coefficients [Jacobs *et al.*1995], or integrate spatial query methods with feature-based methods ([VisualSEEK 1996], [Zabih *et al.*1996], [Stricker *et al.*1996]). Minka

*Research supported in part by ARPA under ONR contract N00014-95-1-0600

and Picard [Minka *et al.*1996] introduced a learning component in their system by using positive and negative examples which lets the system choose image groupings within and across images based on color and texture cues. Their system provides a way to learn labels that describe various parts of the scene based on the examples. Most of these methods described above use absolute properties (e.g. color, texture and spatial information) of the images. If we consider the domain of natural scene classification, these absolute properties can vary between images of the same class. For example, Figure 4 shows variations in color, texture, luminance and shape for images that belong to the class of "Fields". More recently, work by Lipson and Sinha ([Lipson 1996], [Sinha 1994]) in scene classification illustrates that predefined flexible templates that describe the relative color and spatial properties in the image can be used effectively for this task. They demonstrate that the following three ideas are crucial and effective in scene classification: (1) relative relationships matter more than absolute properties, (2) global configuration of the relationships is important and (3) low resolution information is sufficient. In this paper, we focus on automatically learning luminance, color and spatial relations between image patches and build these flexible, relational templates given a set of example images that represent a user-defined class. The system uses a multiresolution approach to extract and refine these relational templates.

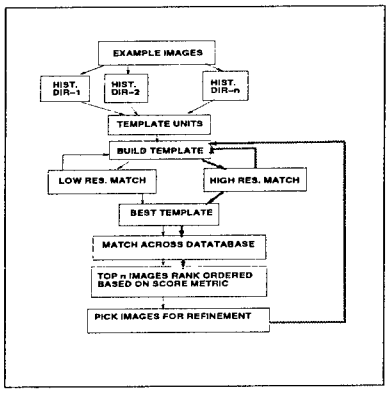


Figure 1: Flow of control in the system

2 Method

Our goal is to construct flexible templates that capture the common spatial, luminance and color patterns across a set of example images. The method used to construct these flexible templates from a training set of images consists of the following steps (Figure 1): building histograms, extracting peaks, building templates, matching process with score metric, incorporating negative examples and refining the template at higher resolution.

Directional histograms of luminance and color relations

Since we want to capture commonality of relationships across a set of example images while allowing for spatial variation, we cannot just use correlation on blurred, low resolution images. We use very low resolution images to build a histogram of relations at the first stage. There are 4 relative relations between a pair of pixels (eg. red, green and blue color channels, luminance). There are 4 choices of symbols for each relationship ($>$, $<$, $=$, $*$) where $*$ is the "don't care symbol", which implies that there are 4^4 bins into which a pair of pixels can cast a vote.

We allow for variation in position as follows. For each pixel in the low resolution image, histogram the signature of the relative relations in a particular direction (Figure 2 LEFT). Build these histograms for different directions separately (e.g. N-S or E-W). We observed that (1) for many classes, the directional histograms had very few bins with multiple entries, and (2) there is a lot of commonality of hits in the histogram across similar images. Each node also has unary color information (eg. node is roughly blue). We ignore the bin which represents the equality relation i.e. regions of uniform color in the image do not contribute to the histogram peaks.

Building templates from histograms

Once we have built the directional histograms,

we extract common peaks in the histograms of the sample images. Each peak gives us a relational signature that defines a part of the final template (Figure 2). Once we pick the top peaks that overlap in the example images, we try combinations of these template units in the direction of the histogram to get possible templates (Figure 2). We match these templates against the example images to pick the ones that best explains the example set. The nodes also have unary color properties. Finally we match the best templates against all the images in the database and rank order the images based on how well they are explained by the templates.

Figure 3 shows examples of the quantized patches in the low resolution images for fields and snowy mountains. The figure also shows the templates that have been extracted from the quantized images using the method described above. These templates capture the color and spatial relations between image patches for these two classes. These extracted templates are then used to classify all the images in the database.

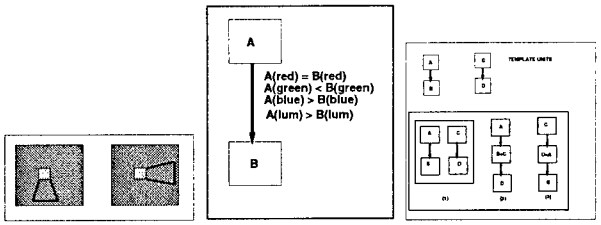


Figure 2: LEFT: Comparisons made to build the directional histograms in the N-S and E-W directions. MIDDLE: Example of a template unit extracted. BOTTOM: Different ways of assembling template units to get the final template that best fits the example images.

2.1 Results using trained templates

We have built an interactive classification system which allows the user to pick a few examples from a random set of images (top two rows of Figure 5). The system extracts the dominant luminance, color and spatial properties of the query images and searches the database for other images that have the same relationships. The system returns an ordered set of images

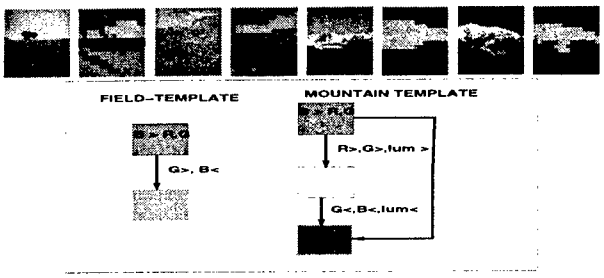


Figure 3: Top Row: Examples of images in fields and snowy-mtns classes and the same low resolution quantized images. Bottom Row: Templates for fields and snowy mountains built from these examples.

that belong the query-class. .

Figure 4 shows the results of the system running with a set of field images as the query and the top matches for the field-template extracted from the example images. These experiments were run on a control set of 800 natural scene images from the Corel collection. Notice that the fields class has images that vary in color, texture and shape and the template needs to be general enough to capture these variations within the class. The last row in the figure shows examples of the false positives in the top 100 matches.

Figure 5 shows the results of training the template on the queries of snowy mountains. In the figure, the query images are the ones selected in red in the top two rows. The figure also shows the top matches returned by the system.

Table 1 summarizes the average performance of the system over 50 trials for two classes (fields and snowy-mountains). There were 110 field images and 100 snowy mountain images in the dataset. These were classified manually for the purpose of evaluating the performance of the system.

3 Summary and Future Directions

In this paper, we have demonstrated a system that classifies images by automatically building flexible templates that capture the luminance, color and spatial relations between image patches, given a set of example images. The system has been tested for a few different classes of

Table 1: Summary of Results using trained templates. The table has the average number of false positives in the top 30 and the top 100 matches and the av. number of false negatives (images in the class that did not get classified correctly by the system) over 50 runs.

Class	FP in top 30	FP in top 100	FN
Fields	5	28	7
Snowy Mountains	3	16	13

natural scenes. The performance in initial test runs indicates that the system can be used to classify images of some natural scenes efficiently and reliably with few explainable false identifications. The system uses a multiple resolution approach where the user has the option of refining the template after the first stage with more specific queries. This method of building relational histograms can also be used to organize the database so that matching at query time can be more effective.

Some of the future directions we are investigating include (1) adding other relational cues (eg. texture) in addition to the color properties described above in order to refine templates and discriminate between images at a finer level, (2) extending the system to include negative examples to reduce the false positive rate and (3) testing and evaluating the system more rigorously with larger databases.

References

[Nayar *et al.*1996] S. Baker and S. Nayar, Pattern Rejection *CVPR*, pp. 544–549, 1996.

[Jacobs *et al.*1995] C.E. Jacobs, A. Finkelstein, and D.H. Salesin, Fast Multiresolution Image Querying *ACM, SIGGRAPH*, pp. 277-286, Los Angeles, CA, 1995.

[Hsu *et al.*1995] W. Hsu, T.S. Chua and H.K. Pung, An integrated color-spatial approach to content-based image retrieval *Proc. ACM Intern. Conf. Multimedia*, pp. 303–313, 1995.

[Lipson 1996] P. Lipson Context and Configuration Based Scene Classification *PhD. Thesis*, MIT, 1996.

[Minka *et al.*1996] T.Minka and R. Picard, Interactive Learning using a "society of models" *CVPR*, 1996.

[Zabih *et al.*1996] G. Pass, R.Zabih, and J. Miller. Comparing images using color coherence vectors. *Proc. ACM Intern. Conf. Multimedia.* Boston, MA, 1996.

[Pentland *et al.*1996] A. Pentland, R. Picard and S. Sclaro, Photobook: Content-based manipulation of image databases *IJCV*, Vol. 18:3, pp. 233–254, 1996.

[QBIC 1995] M. Flickner et al., Query by image and video content: The QBIC System *IEEE Computer*, Vol. 28:9, pp. 23 – 32, 1995.

[Rickman *et al.*1996] R.Rickman and J. Stonham, Content based image retrieval using color-tuple histograms *SPIE Proceedings*, Vol. 2670, pp. 2–7, 1996.

[Sinha 1994] P. Sinha Image Invariants for Object Recognition *Investigative Opthamology and Visual Science*, 1994.

[VisualSEEK 1996] J. Smith and S. Chang. VisualSEEK: a fully automated content-based image query system *Proc. ACM Intern. Conf. Multimedia*, Boston, MA, 1996.

[Stricker *et al.*1996] M. Stricker and A. Dimai, Color indexing with weak spatial contstraints *SPIE Proceedings*, Vol. 2670, pp. 29–40, 1996.

[Swain *et al.*1991] M.J. Swain and D. H.. Ballard, Color Indexing *IJCV*, Vol. 7:1, 1991.

[Virage 1996] J.R. Bach et. al. Virage image search engine: an open framework for image management *Symposium on Electronic Imaging: Science and Technology* , Vol. 4, pp. 76–87, IS&T/SPIE, 1996.

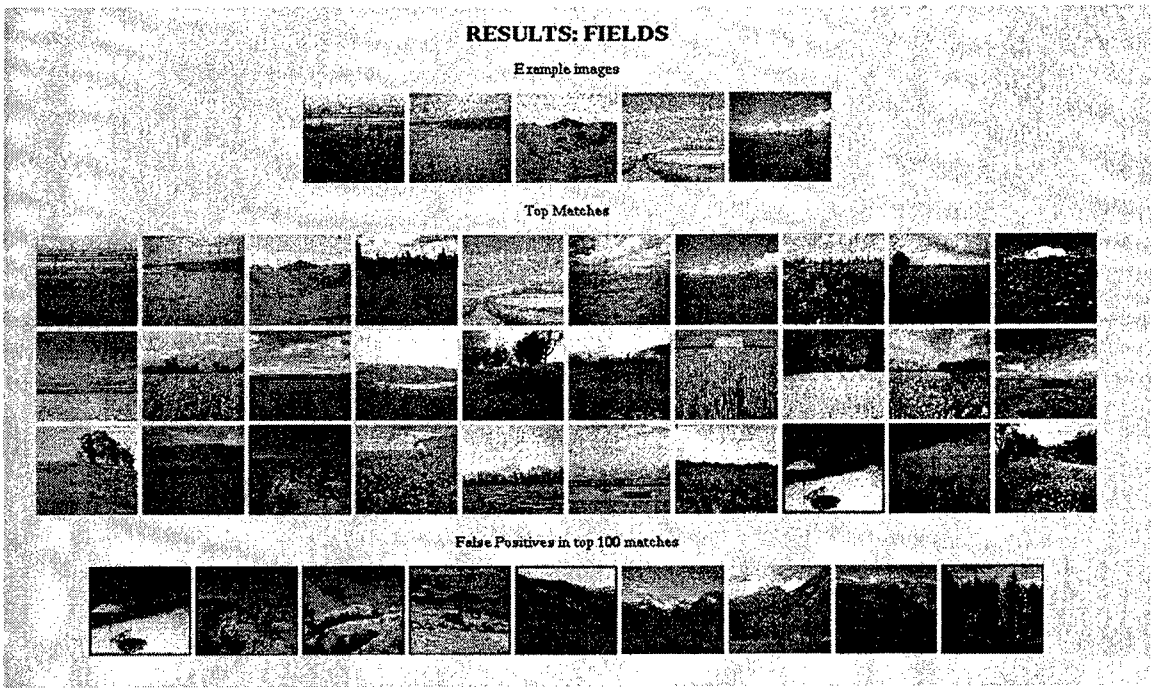


Figure 4: Top Row: Query Images for fields-class. Middle 3 rows: Top 30 matches. Bottom Row: Some of the false positives in the top 100 matches.

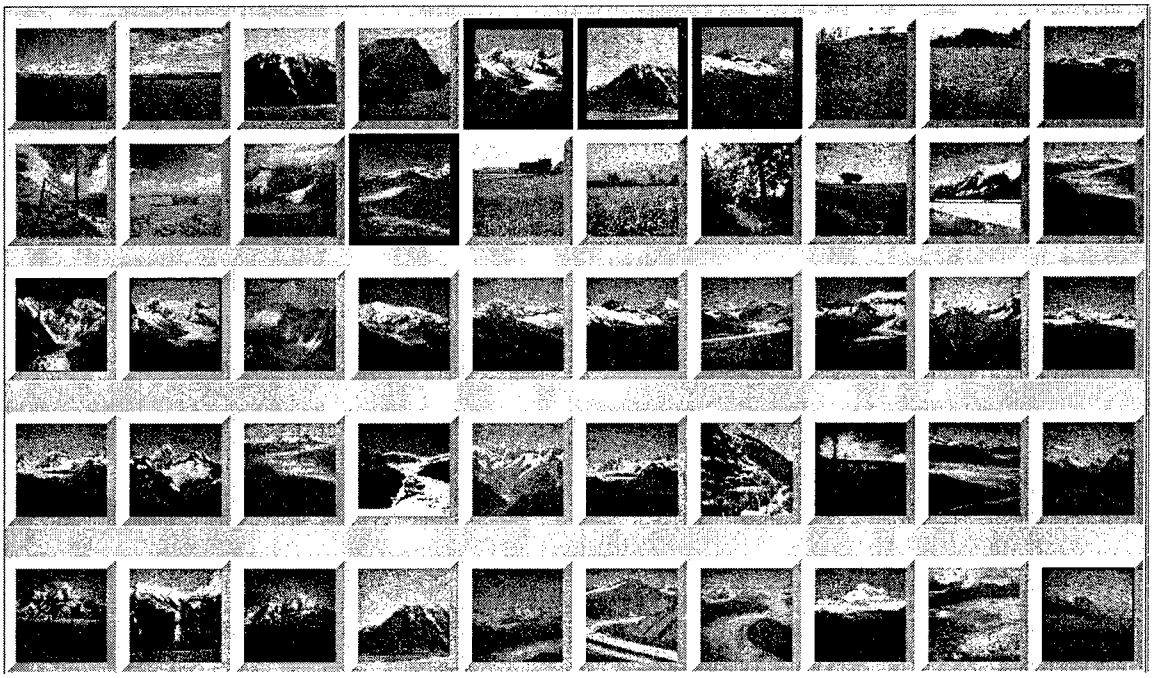


Figure 5: Results of running the system on the examples of snowy mountains selected as the query. TOP TWO ROWS: Random set of images and the query images selected by the user. BOTTOM 3 ROWS: Top 30 matches after extracting the template from the query are shown here.

Combining Color and Spatial Information for Content-based Image Retrieval

Jing Huang Ramin Zabih

Computer Science Department
Cornell University
Ithaca, NY 14853
huang@cs.cornell.edu rdz@cs.cornell.edu

Abstract

Color histograms are widely used for content-based image retrieval, because they are robust to large changes in viewpoint, and can be computed trivially. However, they fail to incorporate spatial information. We have developed several methods for combining color information with spatial layout, while retaining the advantages of histograms. One technique computes the distribution of a given color as a function of the distance between two pixels. The resulting method, which we call a *color correlogram*, has proven to be quite effective even with very coarsely quantized color information. Another method computes joint histograms of local properties, thus dividing pixels into classes based on both color and spatial properties. Experiments demonstrate that these simple, efficient measures perform significantly better than color histograms, especially when the number of images is large.

1 Introduction

Content-based image retrieval requires simple and effective image features for comparing images based on their overall appearance. Color histograms are widely used, for example by QBIC [Flickner *et al.*, 1995], Chabot [Ogle and Stonebraker, 1995] and Photobook [Pentland *et al.*, 1996]. The histogram is easy to compute and is insensitive to small changes in viewing positions. A histogram is a coarse characterization of an image, however, and images with very different appearances can have similar histograms. For example, the images shown in figure 1 have similar color histograms. When image databases are large, this problem is especially acute.

Since histograms do not include any spatial information, recently several approaches have attempted to incorporate spatial information with color [Hsu *et al.*, 1995, Stricker and Dimai, 1996, Smith and Chang, 1996]. These methods, however, lose many of the advantages of color histograms. In this paper we describe methods for combining color information with spatial layout while retaining the advantages of histograms. One method computes the spatial correlation of pairs of colors as a function of the distance between pixels. We call this feature *color correlogram.*[1] Another approach is based on computing joint histograms of several local properties. Joint histograms can be compared as vectors, just as color histograms can. However, in a color histogram any two pixels of the same color are effectively identical. With joint histograms, pixels must share several properties beyond color. We call this approach *histogram refinement.* The methods we describe are easy to compute, and they produce concise summaries of the image.

In sections 2 and 3, we briefly describe color correlograms and histogram refinement (for details see [Huang *et al.*, 1997] and [Pass and Zabih, 1996]). We have evaluated these methods us-

[1] The term "correlogram" is adapted from spatial data analysis [Upton and Fingleton, 1985]

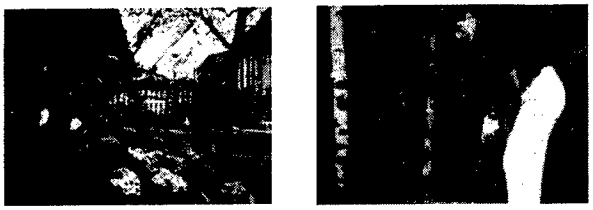


Figure 1: Two images with similar color histograms

ing a large database of images, on tasks with a simple, intuitive notion of ground truth. The experimental results that we present in section 4 show that our methods are significantly more efficient than color histograms.

2 Color Correlograms

A color correlogram (henceforth *correlogram*) expresses how the spatial correlation of pairs of colors changes with distance. Informally, a correlogram for an image is a table indexed by color pairs, where the d-th entry for row $\langle i, j \rangle$ specifies the probability of finding a pixel of color j at a distance d from a pixel of color i in this image. Here d is chosen from a set of distance values D (see [Huang *et al.*, 1997] for the formal definition). An *autocorrelogram* captures spatial correlation between identical colors only. This information is a subset of the correlogram and consists of rows of the form $\langle i, i \rangle$ only.

Since local correlations between colors are more significant than global correlations in an image, a small value of d is sufficient to capture the spatial correlation. We have an efficient algorithm to compute the correlogram when d is small. This computation is linear in the image size ([Huang *et al.*, 1997]).

The highlights of the correlogram method are: (i) it includes the spatial correlation of colors, and (ii) it can be used to describe the global distribution of local spatial correlation of colors if D is chosen to be local (see section 4). An additional advantage lies in the ability of our methods to succeed with very coarse color information. As we show in [Huang *et al.*, 1997], our data suggests that 8-color correlograms perform better than 64-color histograms.

Unlike purely local properties, such as pixel position, gradient direction, or purely global properties, such as color distribution, correlograms take into account the local color spatial correlation as well as the global distribution of this spatial correlation. While any scheme that is based on purely local properties is likely to be sensitive to large appearance changes, (auto)correlograms are more stable to these changes; while any scheme that is based on purely global properties is susceptible to false positive matches, (auto)correlograms prove to be quite effective for content-based image retrieval from a large image database.

3 Histogram Refinement

In *histogram refinement* the pixels of a given bucket are subdivided into classes based on local features. There are many possible features, including texture, orientation, distance from the nearest edge, relative brightness, etc. If we consider color as a random variable, then a color histogram approximates the variable's distribution. Histogram refinement approximates the joint distribution of a variety of local properties.

Histogram refinement prevents pixels in the same bucket from matching each other if they do not fall into the same class. Pixels in the same class can be compared using any standard method for comparing histogram buckets (such as the L_1 distance). This allows fine distinctions that cannot be made with color histograms.

As a simple example of histogram refinement, consider a positional refinement where each pixel in a given color bucket is classified as either "in the center" of the image, or not. Specifically, the centermost 75% of the pixels are defined as the "center". This centering-based refinement produces a split histogram in which the pixels of color buckets are loosely constrained by their location in the image.

3.1 Color coherence vectors

CCV's are a more sophisticated form of histogram refinement, in which histogram buck-

ets are partitioned based on spatial coherence. Our coherence measure classifies pixels as either coherent or incoherent. A coherent pixel is a part of a sizable contiguous region, while an incoherent pixel is not. A *color coherence vector* (or CCV) represents this classification for each color in the image. Sizable contiguous regions can be found by a variety of methods; the one we have used simply computes connected components in the discretized color space and then thresholds the components based on their size. We have also experimented with more complex combinations of local properties; see [Pass and Zabih, 1996] for details. There are two variants of CCV's which we have experimented with. One variant, which we will write as CCV/C, combines coherence with centering-based refinement, thus splitting each bucket into four classes. A particularly useful variant, which we will write as CCV/C/G, includes coherence, centering and the intensity gradient direction.

4 Experimental Results

The methods we describe have been implemented and tested on a large image database. It contains just under 15,000 images, and includes the 11,667 images used in Chabot [Ogle and Stonebraker, 1995], the 1,440 images used in QBIC [Flickner *et al.*, 1995], and a 1,005 image database available from Corel. For some of our tests we also included in the database 40,000 images from CNN taken one minute apart.

Hand examination of our database revealed 77 pairs of images which contain different views of the same scene. One image is selected as a query image, and the other represents a "correct" answer. The queries include various situations like different views of the same scene, large changes in appearance, small lighting changes, spatial translations, etc. In each case, we compute where the second image ranks, when similarity is computed using color histograms or using our methods. The color images shown are available on the web starting at http://www.cs.cornell.edu/home/rdz.

We considered the RGB colorspace with color quantization into 64 colors. We chose the distance set $D = \{1, 3, 5, 7\}$ for computing the autocorrelograms [2]. Since we explored the sparsity of the feature vectors to speed up their processing, the query response time for autocorrelograms is under 2 seconds on a SPARCstation 20. For histogram refinement, we looked at CCV, CCV/C and CCV/C/G.

4.1 Results

Examples of some queries and answers (and the rankings according to the histogram, CCV, CCV/C, and autocorrelogram methods) are shown in Figure 2. As these examples suggest, our methods are robust in tolerating large changes in appearance of the same scene caused by changes in viewing positions, changes in the background scene, partial occlusions, camera zoom that causes radical changes in shape, etc.

To evaluate performance we used two performance measures: *r-measure* which sums up the rank of the correct answers over all queries; p_1-*measure* which sums up (over all queries) the precisions at recall equal to 1. Note that a method is good if it has a low r-measure and a high p_1-measure. Table 1 compares the overall performance of the autocorrelogram, histogram, CCV, and CCV/C using 64 color buckets. The L_1 norm is used to compare feature vectors.

We have also conducted an evaluation of the CCV and CCV/C/G methods on the larger database. Here, we assume the user will examine only the top few answers, and ask the question "how many times does the right answer occur in the top few"? We call the number of images that a user will consider *scope*, and we can evaluate methods by comparing their scope-versus-recall curves. The data shown in table 2 suggests that our methods work significantly better than color histograms.

[2] We did not have to compute the correlogram here as the autocorrelogram itself was sufficient to produce good results.

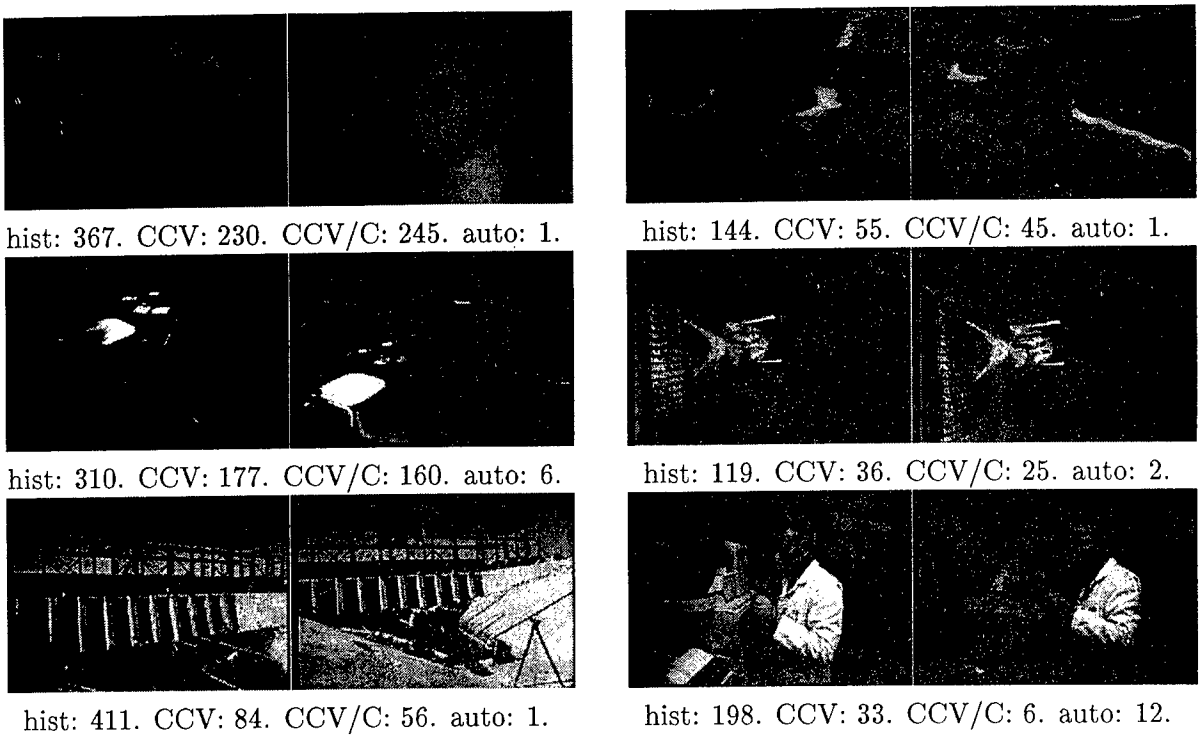


Figure 2: Sample queries and answers with ranks for various methods. Lower ranks indicate better performance. Ranks are from a database of 15,000 images.

Method	hist	CCV	CCV/C	auto
r-measure	6301	3934	3272	172
Average r-measure	82	51	42	2
p_1-measure	21.25	27.54	31.60	58.03
Average p_1-measure	0.28	0.36	0.41	0.75

Table 1: Performance of various methods on 15,000 image database.

4.2 Statistical analysis

We adopt the approach used in [Pass and Zabih, 1996] to analyze the statistical significance of the improvements. We formulate the null hypothesis H_0 which states that the autocorrelogram method is as likely to cause a negative change in rank as a non-negative one. Under H_0, the expected number of negative changes is $M = 38.5$, with a standard deviation $\sigma = \sqrt{77}/2 \approx 4.39$. The actual number of negative changes is 4, which is less than $M - 7\sigma$. We can reject H_0 at more than 99.9% standard significance level.

For histogram refinement, we used a much larger database with a slightly different set of 40 query pairs. In 37 of the 40 cases, CCV's produced better results, while in 1 case they produced worse results and in 2 cases the methods behaved identically. The average change in rank due to CCV's was an improvement of 277 positions (note that this included the 1 case where CCV's did worse, which was a failure of 9 positions). The average percentage change in rank was an improvement of 58%. In the 37 cases where CCV's performed better than color histograms, the average improvement in rank was 300 positions.

The null hypothesis H_0 states that CCV's are equally likely to cause a positive change in ranks (i.e., an improvement) or a negative change. We will discard the two ties to simplify the analysis. Under H_0, the expected number of positive changes is 19, with a standard deviation of

Scope	Hist	CCV	CCV/C/G
1	2.5%	10.0%	47.5%
10	15.0%	42.5%	75.0%
25	32.5%	57.5%	92.5%
50	47.5%	75.0%	92.5%
100	62.5%	77.5%	95.0%
250	65.0%	82.5%	97.5%

Table 2: Recall on 60,000 image database. Higher percentages are better.

$\sqrt{38}/2 \approx 3.08$. The actual number of positive changes is 37, which is almost 6 standard deviations greater than the number expected under H_0. We can therefore reject H_0 at any standard significance level (such as 99.9%).

5 Conclusions

We have demonstrated several ways to combine color and spatial information for image retrieval. These methods incorporate spatial information in different ways, and make different tradeoffs. Images that are similar under one measure may be quite different under another. This in turn suggests that automated methods for combining different measures may be required.

References

[Flickner *et al.*, 1995] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Pater Yanker. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):23–32, September 1995.

[Hsu *et al.*, 1995] Wynne Hsu, T. S. Chua, and H. K. Pung. An integrated color-spatial approach to content-based image retrieval. In *ACM Multimedia Conference*, pages 305–313, 1995.

[Huang *et al.*, 1997] Jing Huang, S. Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Image indexing using color correlograms. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1997. To appear.

[Ogle and Stonebraker, 1995] Virginia Ogle and Michael Stonebraker. Chabot: Retrieval from a relational database of images. *IEEE Computer*, 28(9):40–48, September 1995.

[Pass and Zabih, 1996] Greg Pass and Ramin Zabih. Histogram refinement for content-based image retrieval. In *IEEE Workshop on Applications of Computer Vision*, pages 96–102, December 1996.

[Pentland *et al.*, 1996] Alex Pentland, Rosalind Picard, and Stan Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, June 1996.

[Smith and Chang, 1996] J. R. Smith and S.-F. Chang. VisualSEEk: A fully automated content-based image query system. In *ACM Multimedia Conference*, November 1996.

[Stricker and Dimai, 1996] Markus Stricker and Alexander Dimai. Color indexing with weak spatial constraints. *SPIE proceedings*, 2670:29–40, February 1996.

[Upton and Fingleton, 1985] Graham J. G. Upton and Bernard Fingleton. *Spatial Data Analysis by Example*, volume I. John Wiley & Sons, 1985.

A Characterization of Visual Appearance Applied to Image Retrieval*

S. Ravela and **R. Manmatha**
Multimedia Indexing and Retrieval Group
Center for Intelligent Information Retrieval & Computer Vision Laboratory
University of Massachusetts at Amherst
{ravela,manmatha}@cs.umass.edu
URL: http://vis-www.cs.umass.edu/ ravela

Abstract

A system to retrieve images using a description of visual appearance is presented. A multi-scale invariant vector representation is obtained by first filtering images in the database with Gaussian derivative filters at several scales and then computing low order differential invariants. The multi-scale representation is indexed for rapid retrieval. Queries are designed by the users from an example image by selecting appropriate regions. The invariant vectors corresponding to these regions are matched with those in the database both in feature space as well as in coordinate space and a match score is obtained for each image. The results are then displayed to the user sorted by the match score. From experiments conducted with over 1500 images it is shown that images similar in appearance and whose viewpoint is within 25 degrees of the query image can be retrieved with a very satisfactory average precision[1] of 57.4%

1 Introduction

The goal of image retrieval systems is to operate on collections of images and, in response to visual queries, extract relevant images. The application potential for fast and effective image retrieval is enormous, ranging from database management in museums and medicine, architecture and interior design, image archiving, to constructing multi-media documents or presentations[4]. However, there are several issues that must be understood before image retrieval can be successful. Foremost among these is an understanding of what 'retrieval of relevant images' means. Relevance, for users of a retrieval system, is most likely associated with semantics. Encoding semantic information into a general image retrieval system entails solving such problems as feature extraction, segmentation and, object and context recognition. These are extremely hard problems that are as yet unsolved. However, in many situations attributes associated with an image, when used together with some level of user input, correlate well with the kind of semantics that are desirable. Consequently, recent work has focused directly on surface level image content descriptions such as color[20], texture features [10, 3, 14, 11], shape [12, 24] and combinations thereof [1, 5, 14].

In this paper images are retrieved using a characterization of the visual appearance of objects. An object's visual appearance in an image depends not only on its three-dimensional geometric shape, but also on its albedo, its surface texture, the view point from which it is imaged, among other factors. It is non-trivial to separate the different factors that constitute an object's visual appearance. However, we posit that the shape of an imaged object's intensity surface closely relates to its visual appearance. Here a local characterization of the intensity surface is constructed and images are retrieved using a measure of similarity for this representation. The experiments conducted in this paper verify the association that objects that appear to be visually similar can be retrieved by a characterization of the shape of the intensity surface.

Different representations of appearance have been used in object recognition [13, 18] and have been applied to specific types of retrieval such as face recognition [6, 23]. To the best of our knowledge the system presented here is the first attempt to characterize appearance to retrieve similar images and in this paper the development of Synapse (Syntactic Appearance Search Engine), an image database

This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, in part by the United States Patent and Trademark Office and Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235, in part by the National Science Foundation, Central Intelligence Agency, Department of Defense (DARPA) and National Security Agency under grant number IRI-9619117 and in part by NSF Multimedia CDA-9502639. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsors.

[1] precision is the proportion of retrieved images that are relevant

search engine, is described. The approach taken here does not rely on image segmentation (manual or automatic) or binary feature extraction. Unlike some of the previously mentioned methods, no training is required. Since the representation is local, objects can be embedded in different backgrounds. Using an *example image and user interaction to construct queries*, Synapse retrieves similar images within small view and size variation in the order of their *similarity in syntactic appearance to a query.*

The claim is that, up to a certain order, the *local appearance* of the intensity surface (around some point) can be represented as responses to a set of scale parameterized Gaussian derivative filters (see Section 3). This set or vector of responses, called a multi-scale feature vector, is obtained solely from the signal content and without the use of "global context" or "symbolic interpretation". Further, the family of Gaussian filters are unique in their ability to describe the *scale-space* or *deep structure* [7, 9, 22, 2] of a function and are well suited for representing appearance.

In this paper an indexable strategy for image retrieval is developed using feature vectors constructed from combinations of the derivative filter outputs. These combinations yield a set of differential invariants [2] that are invariant to two-dimensional rigid transformations. Retrieval is achieved in two computational steps. During the off-line computation phase each image in the database is first filtered at sampled locations and then filter responses across the entire database are indexed(see Section 3). The run-time computation of the system begins with the user selecting an example image and marking a set of salient regions within the image. The responses corresponding to these regions are matched with those of the database and a measure of fitness per image in the database is computed in both feature space and coordinate space (see Section 4). Finally, images are displayed to the user in the order of fitness (or match score) to the query (see Section 5).

2 Related Work

Eigen-space representations [13, 6, 23, 21] are one of the earliest attempts to characterize appearance or the intensity shape. This space is constructed by treating the image as a fixed length vector, and then computing the principal components across the entire database. The images therefore have to be size and intensity normalized, segmented and involves training. The approach presented in this paper does not characterize appearance by eigen decomposition or any variation thereof. Further, the method presented uses no learning, does not depend on constant sized images, tolerates significant variation in background and retrieves from heterogeneous collections of images using local representations of appearance.

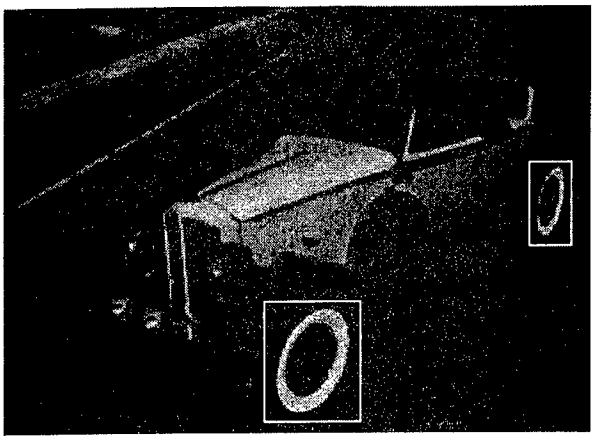


Figure 1: Allowing the user to construct queries by selecting the box shown

Gaussian derivative representations have been used in the context of recognition [15]. Indexed differential invariants have recently been used [18] for object recognition. We also index on differential invariants but there are several differences. First, the invariants corresponding to the low two order derivatives are used (as opposed to the first nine invariants), for reasons of speed as well as relevance to retrieving similar images(see section 3). Second, their indexing algorithm depends on interest point detection and is, therefore, limited by the stability of the interest operator. We on the other hand sample the image. Third, the authors do not incorporate multiple scales into a single vector whereas here three different scales are chosen. In addition the index structure and spatial checking algorithms differ.

The earliest general image retrieval systems were designed by [1, 14]. In [1] the shape queries require prior manual segmentation of the database which is undesirable and not cost-effective for most applications. Texture based image retrieval is also related to the appearance based work presented in this paper. Using Wold modeling[10] , the authors try to classify the entire Brodatz texture set and in [3] they attempt to classify scenes, such as city and country. Of particular interest is work by [11] who use Gabor filter representation (globally over the entire image) to retrieve by texture similarity.

3 Syntactic Representation of Appearance

This section begins by making explicit the notion of appearance and the uniqueness of Gaussian derivative filters therein. Then a representation, namely a multi-scale feature vector is constructed by filtering an image with a set of Gaussian derivative filters. The multi-scale feature vector are transformed so that the elements within this vector are invariant to 2D rigid transformations. This transformed fea-

ture vector is called the multi-scale invariant vector. Then a scheme for indexing multi-scale invariant vectors computed over the entire image database is presented. This completes all the steps of the off-line computation described earlier.

3.1 Characterization of Appearance

A function can be locally characterized by its Taylor series expansion provided the derivatives at the point of expansion are well conditioned. The intensity function of the image, on the other hand, need not satisfy this condition. However, it is well known that the derivative of a possibly discontinuous function can be made well posed if it is convolved with the derivative of a smooth test function [19]. Consider the normalized Gaussian as a choice for the smooth test function. Then the derivatives of the image $I_\sigma(\mathbf{x}) = (I \star G)(\mathbf{x}, \sigma), \mathbf{x} \in \Re^2, \sigma \in \Re^+$, are well conditioned for some value of σ. This is written as

$$I_{i_1 \ldots i_n, \sigma}(\mathbf{x}) = (I \star G_{i_1 \ldots i_n})(\mathbf{x}, \sigma)$$

$$G_{i_1 \ldots i_n} = \frac{\delta^n}{\delta_{i_1} \ldots \delta_{i_n}} G$$

and $i_k = x_1 \ldots x_D, k = 1 \ldots n$.

The local N-jet of $I(\mathbf{x})$ at scale σ and order N is defined as the set [8]:

$$J^N[I](\mathbf{x},\sigma) = \{I_{i_1 \ldots i_n, \sigma} | n = 0 \ldots N\} \quad (1)$$

It
can be observed that the set $\lim_{N \to \infty} J^N[I](\mathbf{x}, \sigma)$ bundles all the derivatives required to fully specify the Taylor expansion of I_σ up to derivatives of order N. Thus, for any order N, the local N-jet at scale σ locally contains all the information required to reconstruct I at the scale of observation σ up to order N. This is the primary observation that is used to characterize appearance. That is, up to any order the derivatives locally characterize the shape of the intensity surface, i.e. appearance, to that order. From the experiments shown in this paper it is also observed that this representation can be used to retrieve images that appear visually similar.

The choice of the Gaussian as the smooth test function, as opposed to others, is motivated by the fact that it is unique in describing the scale-space or deep structure of an arbitrary function. A full review of scale-space is beyond the scope of this paper and the reader is referred to [26, 7, 2, 9, 22]. Here some of the important consequences of incorporating scale space are considered. For increasing values of σ the Gaussian filter admits a narrowing band of frequencies and I will appear smoother. The scale-space of I is simply I_σ, where σ is the free variable. Similarly, the scale space of the derivatives of I is the range of $I_{i_1 \ldots i_n, \sigma}$ where σ is the free variable. Scale-space has an important physical interpretation in that it models the change in appearance of an imaged object as it moves away from a camera. An argument is therefore made for a *multi-scale feature vector* which describes the intensity surface locally at several scales. From an implementation stand point a *multi-scale feature vector* at a point p in an image I is simply the elements of the vector:

$$\{J^N[I](\mathbf{p}, \sigma_1), J^N[I](\mathbf{p}, \sigma_2) \ldots J^N[I](\mathbf{p}, \sigma_k)\} \quad (2)$$

for some order N and a set of scales $\sigma_1 \ldots \sigma_k$. In practice the zeroth order terms are dropped to achieve invariance to constant intensity changes. Multi-scale vectors represent appearance more robustly than a single-scale vector. This can viewed from several different perspectives. Since, multi-scale vectors are values computed at several different kernel sizes, therefore, they contain more information than fixed window operators. Equivalently, multi-scale vectors contain information at several different bandwidths and with the choice of a Gaussian accurately represent the intensity shape at different depths from the camera. From a practical standpoint this means that mis-matches due to an accidental similarity at a single scale can be reduced.

A measure of similarity between two multi-scale vectors can be obtained by correlating them or computing the distance between the vectors. In earlier work [17] it was shown that multi-scale vectors can be used to retrieve images visually similar and within a small view and finite scale variation of the query. An important observation from that work is that as images become more dissimilar their response vectors become less correlated, starting at the higher order. Thus, similar images can be expected to be more correlated in their lower order than higher ones. Similar arguments can be made for scales. As images get dissimilar, they can be expected to retain strong correlation only at large scales (lower spatial frequency). Further the range of scales over which they correlate well gets smaller. As a consequence, in this paper the multi-scale vector is computed at three different scales placed half an octave apart. This is discussed in the next subsection.

3.2 Multi-Scale Invariant Vectors

The limitation of using the derivatives directly in a feature vector is that it has restricted tolerance to rotations. This issue is partially addressed by transforming the multi-scale feature vector so that it is invariant to 2D rigid transformations.

Given the derivatives of an image I, *irreducible differential invariants* (invariant under the group of displacements) can be computed in a systematic manner [2]. The term irreducible is used because other invariants can be reduced to a combination of the irreducible set. The value of these entities is independent of the choice of coordinate frame (up to rotations) and the terms for the low orders (two here) are enumerated below.

The irreducible set of invariants up to order two of an image I are:

$$
\begin{array}{lll}
d_0 & = I & \text{Intensity} \\
d_1 & = I_x^2 + I_y^2 & \text{Magnitude} \\
d_2 & = I_{xx} + I_{yy} & \text{Laplacian} \\
d_3 & = I_{xx} I_x I_x + 2 I_{xy} I_x I_y + I_{yy} I_y I_y & \\
d_4 & = I_{xx}^2 + 2 I_{xy}^2 + I_{yy}^2 &
\end{array}
$$

In experiments conducted in this paper, the vector, $\Delta_\sigma = \langle d_1, \ldots d_4 \rangle_\sigma$ is computed at three different scales. The element d_0 is not used since it is sensitive to gray-level shifts. The resulting multi-scale invariant vector has at most twelve elements. Computationally, each image in the database is filtered with the first five partial derivatives of the Gaussian (i.e. to order 2) at three different scales at uniformly sampled locations. Then the multi-scale invariant vector $D = \langle \Delta_{\sigma_1}, \Delta_{\sigma_2}, \Delta_{\sigma_3} \rangle$ is computed at those locations.

A location across the entire database can be identified by the *generalized coordinates*, defined as, $c = (i, x, y)$ where i is the image number and (x, y) a coordinate within this image. The computation described above generates an association between generalized coordinates and invariant vectors. This association can be viewed as a table $M : (i, x, y, D)$ with $3 + k$ columns(k is the number of fields in an invariant vector) and number of rows, R, equal to the total number of locations (across all images) where invariant vectors are computed.

To retrieve images, a 'find by value' functionality is needed, with which, a query invariant vector is found within M and the corresponding generalized coordinate is returned. The brute force approach entails a linear search in M which is extremely time consuming. The solution is to generate inverted files (or tables) for M, based on each field of the invariant vector and index them. Then the operation of 'find-by-value' can be performed in $\log(R)$ time (number of rows) and is described below.

To index the database by fields of the invariant vector, the table M is split into k smaller tables $M_1' \ldots M_k'$, one for each of the k fields of the invariant vector. Each of the smaller tables $M_p', p = 1 \cdots k$ contains the four columns $(D(p), i, x, y)$. At this stage any given row across all the smaller tables contains the same generalized coordinate entries as in M. Then, each M_p' is sorted and a binary tree is used to represent the sorted keys. As a result, the entire database is indexed.

4 Matching Invariant Vectors

Run-time computation begins with the user marking selected regions in an example image. At sampled locations within these regions, invariant vectors are computed and submitted as a query. The search for matching images is performed in two stages. In the first stage each query invariant is supplied to the 'find-by-value' algorithm and a list of matching generalized coordinates is obtained. In the second stage a spatial check is performed on a per image basis, in order to verify that the matched locations in an image are in spatial coherence with the corresponding query points. In this section the 'find-by-value' and spatial checking components are discussed.

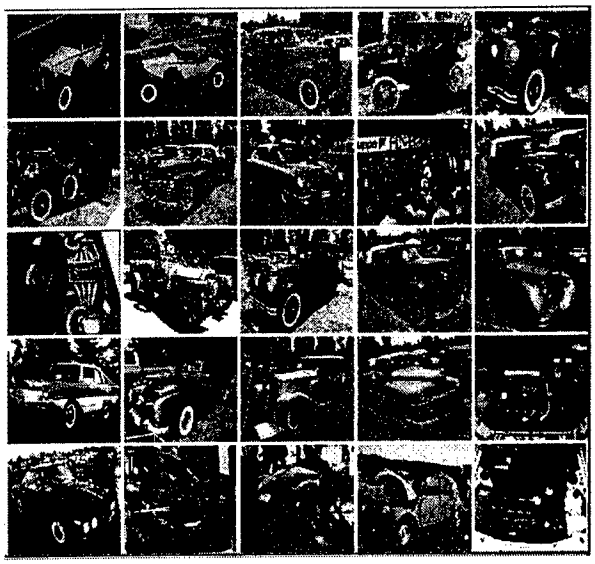


Figure 2: The results of the car query shown in Figure 1

4.1 Finding by Invariant Value

The multi-scale invariant vectors at sampled locations within regions of a query image can be treated as a list. The n^{th} element in this list contains the information $Q_n = (D_n, x_n, y_n)$, that is, the invariant vector and the corresponding coordinates. In order to find-by-invariant-value, for any query entry Q_n, the database must contain vectors that are within a threshold $t = (t_1 \ldots t_k) > 0$. The coordinates of these matching vectors are then returned. This can be represented as follows. Let p be any invariant vector stored in the database. Then p matches the query invariant entry D_n only if $D_n - t < p < D_n + t$. This can be rewritten as

$$\&_{j=1}^{k} \left[D_n(j) - t(j) < p(i) < D_n(j) - t(j) \right]$$

where & is the logical *and* operator and k is the number of fields in the invariant vector. To implement the comparison operation two searches can be performed on each field. The first is a search for the lower bound, that is the largest entry smaller than $D_n(j) - t(j)$ and then a search for the upper-bound i.e. the smallest entry larger than $D_n(j) + t(j)$. The block of entries between these two bounds are those that match the field j. In the inverted file the generalized coordinates are stored along with the individual field values and the block of matching generalized coordinates are copied from disk. To implement the logical-and part, an intersection

of all the returned block of generalized coordinates is performed. The generalized coordinates common to all the k fields are the ones that match query entry Q_n. The find by value routine is executed for each Q_n and as a result each query entry is associated with a list of generalized coordinates that it matches.

4.2 Spatial-Fitting

The association between a query entry Q_n and the list of f generalized coordinates that match it by value can be written as

$$\begin{aligned} A_n &= \langle x_n, y_n, c_{n_1}, c_{n_2} \ldots c_{n_f} \rangle \\ &= \langle x_n, y_n, (i_{n_1}, x_{n_1}, y_{n_1}) \ldots (i_{n_f}, x_{n_f}, y_{n_f}) \rangle \end{aligned}$$

Here x_n, y_n are the coordinates of the query entry Q_n and $c_{n_1} \ldots c_{n_f}$ are the f matching generalized coordinates. The notation c_{n_f} implies that the generalized coordinate c matches n and is the f^{th} entry in the list. Once these associations are available, a spatial fit on a per image basis can be performed. In order to describe the fitness measure, two definitions are needed. First, define the distance between the coordinates of two query entries m and n as $\delta_{m,n}$. Second, define the distance between any two generalized coordinates c_{m_j} and c_{n_j} that are associated with two query entries m, n as $\delta_{c_{m_j}, c_{n_k}}$

Any image u that contains two points (locations) which match some query entry m and n respectively are coherent with the query entries m and n only if the distance between these two points is the same as the distance between the query entries that they match. Using this as a basis, a binary fitness measure can be defined as

$$\mathcal{F}_{m,n}(u) = \begin{cases} 1 & \text{if } \exists j \exists k \mid \left| \delta_{m,n} - \delta_{c_{m_j}, c_{n_k}} \right| \leq T \\ & i_{m_j} = i_{n_k} = u, m \neq n \\ 0 & \text{otherwise} \end{cases}$$

That is, if the distance between two matched points in an image is close to the distance between the query points that they are associated with, then these points are spatially coherent (with the query). Using this fitness measure a match score for each image can be determined. This match score is simply the maximum number of points that together are spatially coherent (with the query). Define the match score by:

$$score(u) \equiv \overset{max}{m} S_m(u) \qquad (3)$$

where, $S_m(u) = \sum_{n=1}^{f} \mathcal{F}(u)_{m,n}$. The computation of $score(u)$ is at worst quadratic in the total number of query points. The array of scores for all images is sorted and the images are displayed in the order of their score. T used in $\mathcal{F}$ is a threshold and is typically 25% of $\delta_{m,n}$. Note that this measure not only will admit points that are rotated but will also tolerate other deformations as permitted by the threshold. The value of the threshold is selected to reflect the rationale that similar images will have similar responses but not necessarily under a rigid deformation of the query points.

4.3 Query Construction

The ability for the user to construct queries by selecting regions is an important distinction between the approach presented here and elsewhere. Users can be expected to employ their considerable semantic knowledge about the world to construct a query. Such semantic information is difficult to incorporate in a system. An example of query construction is shown in Figure 1, where the user has decided to find cars similar to the one shown and decides that the most salient part are 'wheels'[2]. It is clear that providing such interaction removes the necessity for automatic determination of saliency. In the car example, the user provides the context to search the database by marking the wheel and retrieved images mostly contain wheels. The association of wheels to cars is not known to the system, rather it is one that the user decides is meaningful. Several other approaches in the literature take the entire feature set or some global representation over the entire image[1, 4, 21, 11]. While this may be reasonable for certain types of retrieval, it cannot necessarily be used for general purpose retrieval.Therefore, we believe that the natural human ability in selecting salient regions must be exploited. More importantly, letting the user design queries eliminates the need for detecting the salient portions of an object, and the retrieval can be customized so as to remove unwanted portions of the image. Based on the feedback provided by the results of a query, the user can quickly adapt and modify the query to improve performance.

5 Experiments

The database used in this paper has digitized images of cars, steam locomotives, diesel locomotives, apes, faces, people embedded in different background(s) and a small number of other miscellaneous objects such as houses. 1561 images were obtained from the Internet and the Corel photo-cd collection to construct this database. These photographs were taken with several different cameras of unknown parameters, and under varying uncontrolled lighting and viewing geometry. Also, the objects of interest are embedded in natural scenes such as car shows, railroad stations, country sides and so on. The choice of images reflects two primary considerations. First, the images should not reflect a bias towards any particular attribute and second, the system must be able to rank dissimilar images with little difficulty. This is confirmed by the experiments performed to date.

[2]see Figure 2 for the results

Table 1: Queries submitted to the system and expected retrieval

Given(User Input)	Find	Precision
wheel, Figure 1	White Wheeled Cars, see Figure 2	57.0% (see text)
wheel (front wheel only)	White Wheeled Cars	48.6%
Monkey's coat	Dark Textured Apes	57.5%
Face	All Faces	74.7%
Face	Same Person's Face	61.7%
Patas Monkey Face	All Visible Patas Monkey Faces	44.5%

A measure of the performance of the retrieval engine can be obtained by examining the recall/precision table for several queries. Briefly, recall is the proportion of the relevant material actually retrieved and precision is the proportion of retrieved material that is relevant [25]. Consider as an example the query described in Figure 1. Here the user wishes to retrieve 'white wheel cars' similar to the one outlined and submits the query. The query has both wheels marked to impose additional spatial constraints. The top 25 results ranked in text book fashion are shown in Figure 2. Note that although there are several valid matches as far as the algorithm is concerned (for example image 8 a train), they are not considered valid retrievals as stated by the user and are not used in measuring the recall/precision. This is inherently a conservative estimate of the performance of the system. The average precision (over recall intervals of 10^3) is 48.6%. Five other queries that were also submitted are depicted in table 1. Due to lack of space detailed explanations are not provided and the reader is referred to [16] for details. The recall/precision table over these five queries is in Table 2. The average precision over all the queries is a 57.4%. This compares well with text retrieval where some of the best systems have an average precision of 50%[4].

Unsatisfactory retrieval occurs for several reasons. First it is possible that the query is poorly designed. In this case the user can design a new query and re-submit. Also Synapse allows users to drop any of the displayed results into a query box and resubmit. Therefore, the user can not only redesign queries on the original image, but also can use any of the result pictures to refine the search. A second source of error is in matching generalized coordinates by value. The choice of scales in the experiments carried out in this case are $\frac{3}{\sqrt{2}}, 3, \frac{3}{\sqrt{2}}$. It is possible that locally the intensity surface may have a very close value, so as to lie within the chosen threshold and thus introduce an incorrect point. By adding more scales or derivatives such errors can be reduced, but at the cost of increased discrimination and decreased generalization. Many of these 'false matches' are eliminated in the spatial checking phase. Errors can also occur in the spatial checking phase because it admits much more than a rotational transformation of points with respect to the query configuration. Overall the performance to date has been very satisfactory and we believe that by experimentally evaluating each phase the system can be further improved.

The time it takes to retrieve images is dependent linearly on the number of query points. On a Pentium Pro-200 Mhz Linux machine, typical queries execute in between one and six minutes.

6 Conclusions, Limitations and Future Work

Within small view variations, images that are similar to a query are retrieved. These images are also observed to be visually similar and we posit that this method has good potential for image retrieval.

While a discussion of matching objects across different sizes was presented and has been implemented elsewhere [17], in this paper, the multi-scale invariant vector was used only to robustly characterize appearance. The next immediate step is to explicitly incorporate matching across size variations.

A second important question is, what types of invariants should constitute a feature vector ? This is an open research issue. Finally, although the current system is some what slow, it is yet a remarkable improvement over our previous work. We believe that by examining the spatial checking and sampling components further increases in speed are possible.

Acknowledgements

The authors wish to thank Adam Jenkins and Morris Hirsch for programming support and Prof. Bruce Croft and CIIR for continued support of this work.

References

[1] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Denis Lee, Dragutin Petkovix, Devid Steele, and Peter Yanker. Query by image and video content: The qbic system. *IEEE Computer Magazine*, 28(9):23–30, September 1995.

[3] The value $n(= 10)$ is simply the retrievals up to recall n.

[4] Based on personal communication with Bruce Croft

Table 2: Precision at standard recall points for Five Queries

Recall	0	10	20	30	40	50	60	70	80	90	100
Precision %	100	95.1	88.7	76.8	61.8	55.3	44.8	38.5	34.8	21.0	14.2
				average	57.4%						

[2] Ludvicus Maria Jozef Florack. *The Syntactic Structure of Scalar Images.* PhD thesis, University of Utrecht, 1993.

[3] M. M. Gorkani and R. W. Picard. Texture orientation for sorting photos 'at a glance'. In *Proc. 12th Int. Conf. on Pattern Recognition*, pages A459–A464, October 1994.

[4] Venkat N. Gudivada and Vijay V. Raghavan. Content-based image retrieval systems. *IEEE Computer Magazine*, 28(9):18–21, September 1995.

[5] A. K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29:1233–1244, 1996.

[6] M Kirby and L Sirovich. Application of the kruhnen-loeve procedure for the characterization of human faces. *IEEE Trans. Patt. Anal. and Mach. Intel.*, 12(1):103–108, January 1990.

[7] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–396, 1984.

[8] J. J Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[9] Tony Lindeberg. *Scale-Space Theroy in Computer Vision.* Kluwer Academic Publishers, 1994.

[10] Fang Liu and Rosalind W Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Trans. PAMI*, 18(7):722–733, July 1996.

[11] W. Y. Ma and B. S. Manjunath. Texture-based pattern retrieval from image databases. *Multimedia Tools and Applications*, 2(1):35–51, January 1996.

[12] Rajiv Mehrotra and James E. Gary. Similar-shape retrieval in shape data management. *IEEE Computer*, 28(9):57–62, September 1995.

[13] S. K. Nayar, H. Murase, and S. A. Nene. Parametric appearance representation. In *Early Visual Learning.* Oxford University Press, February 1996.

[14] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of databases. In *Proc. Storafe and Retrieval of Image and Video Databases II*, volume 2, pages 34–47. SPIE, 1994.

[15] Rajesh Rao and Dana Ballard. Object indexing using an iconic sparse distributed memory. In *Proc. International Conference on Computer Vision*, pages 24–31. IEEE, 1995.

[16] S. Ravela and R. Manmatha. Image retrieval by appearance. In *(submitted) SIGIR*, 1997.

[17] S. Ravela, R. Manmatha, and E. M. Riseman. Image retrieval using scale-space matching. In Bernard Buxton and Roberto Cipolla, editors, *Computer Vision - ECCV '96*, volume 1 of *Lecture Notes in Computer Science*, Cambridge, U.K., April 1996. 4th European Conf. Computer Vision, Springer.

[18] Cordelia Schmid and Roger Mohr. Combining greyvalue invariants with local constraints for object recognition. In *Proc. Computer Vision and Pattern Recognition.* IEEE, June 1996.

[19] L. Schwartz. Théorie des distributions. In *Actualités scientifiques et industrielles*, volume I,II, pages 1091–1122. Publications de l'Institut de Mathématique de l'University de Strasbourg, 1950-51.

[20] M. Strickler and M. Orengo. Similarity of color images. In *Storage and Retrieval for Image and Video Databases III*, volume 2420 of *SPIE Proceedings Series*, pages 318–192, 1995.

[21] D. L. Swets and J. Weng. Using discriminant eigen features for retrieval. *IEEE Trans. Patt. Anal. and Mach. Intel.*, 18:831–836, August 1996.

[22] Bart M. ter Har Romeny. *Geometry Driven Diffusion in Computer Vision.* Kluwer Academic Publishers, 1994.

[23] M. Turk and A. Pentland. Eigen faces for recognition. *Jrnl. Cognitive Neuroscience*, 3:71–86, 1991.

[24] A. Vailaya, Y. Zhong, and A. K. Jain. A hierarchical system for efficient image retrieval. In *Proc. Int. Conf. on Patt. Recog.*, August 1996.

[25] C. J. van Rijsbergen. *Information Retrieval.* Butterworths, 1979.

[26] A. P. Witkin. Scale-space filtering. In *Proc. Intl. Joint Conf. Art. Intell.*, pages 1019–1023, 1983.

Feature Selection for Robust Color Image Retrieval

Madirakshi Das and Edward M. Riseman*
Multimedia Indexing and Retrieval Group
Joint group of Center for Intelligent Information Retrieval and Computer Vision laboratories
Department of Computer Science
University of Massachusetts, Amherst, MA 01003-4610
E-MAIL: mdas,riseman@cs.umass.edu
HOMEPAGE: http://vis-www.cs.umass.edu/~mdas

Abstract

This work addresses the issue of color feature selection for content-based retrieval from large, heterogeneous color image databases where no assumptions can be made about the images or the type of queries. The color features used to describe an image have been developed based on the need for speed in matching and ease of computation on complex images while maintaining invariance to differences in scale, orientation, and location of the queried object in the database images and also the presence of significant, interfering backgrounds. The colors present and their spatial relationships are used as features to describe a color image. These features are used in an efficient, multi-phase retrieval system to produce retrieval results fast enough for use with an online user. Test results with multi-colored query objects from man-made and natural domains highlight the capabilities of the system.

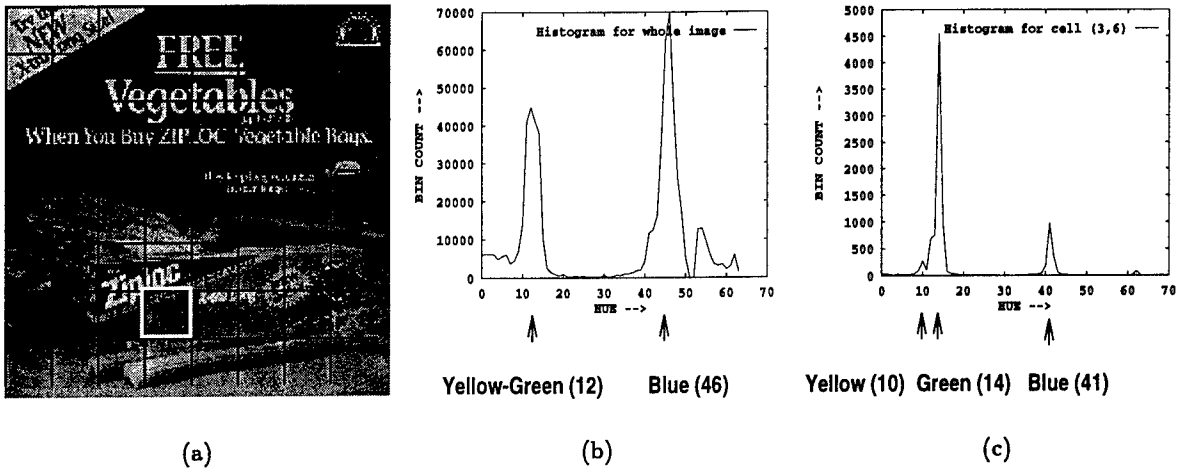
*This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, in part by the United States Patent and Trademark Office and Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235, in part by the National Science Foundation, Central Intelligence Agency, Department of Defense (DARPA) and National Security Agency under grant number IRI-9619117, in part by ARPA (via USAF Rome Laboratory) under contract F30602-94-C-0042 and in part by NSF Multimedia CDA-9502639. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsors.

1 Introduction

With the growing number of multimedia databases, retrieval of images, audio and video from large databases has become an active area of research. As in text retrieval, objects in the database which are relevant to the query being posed by the user need to be retrieved. Content-based retrieval is a popular paradigm for ensuring relevance in image retrieval where the aim is to find the images in a database which contain the object represented in a query image.

The fact that there are no obvious features across images which carry semantic information like words do in text, makes the selection of descriptive features for an image difficult. However, when the database has images of multi-colored objects which can be recognized on the basis of their distinctive color signatures, the color of the object and related color-based features are an obvious choice for indexing.

There has been work in color-based retrieval using color histograms [Swain and Ballard, 1991][Hafner *et al.*, 1995], but the retrieval results are sensitive to difference in scale and viewpoint between the object as depicted in the query image and as present in the database images. Using color clusters [Kankanhalli *et al.*, 1996] avoids the scale problem but both strategies are affected by the presence of interfering backgrounds, particularly when the query image is a small embedded part in a large target image. We have considered the general case where multi-colored objects occur with sig-



(a) (b) (c)

Figure 1: Effect of interfering background on histogram peak location: (a) "Ziploc" advertisement with a cell highlighted (b) global hue histogram of whole image with relevant peaks labelled (c) hue histogram of marked cell with peaks labelled

nificant, interfering backgrounds and in widely varying sizes, locations and orientation in the database images.

2 Selection of features

The main requirement for the color characteristics selected for matching is to provide discrimination between images which contain objects similar to the query object and those which do not. The feature matched needs to be invariant to differences in the scale, location and orientation of the query object in the candidate image and the presence of background colors in the candidate image. It is also desirable for the characteristics to be indexable and the matching process to be fast.

We have used two scale and orientation invariant color features, describing the *color content* of an image and the *spatial relationships* between color regions. In general, there is a trade-off between the discriminatory power of a color feature and its speed of matching. Simpler features are easy for indexing and matching, while complex features which provide more discriminatory power may not be indexable and take longer to match. We have selected both types of features and employed a two phase matching strategy to balance the trade-off between speed of retrieval and the precision obtained. The emphasis in the first phase of matching is on speed of retrieval, and the second phase aims at removal of false matches from the image list produced by the first phase.

2.1 Histogram Peaks as features

The simplest constraint on a database image retrieved as a response to a query is that it must have all the colors of the query object present. To check for this requirement, we need to describe the *color content* of an image. As observed in [Matas *et al.*, 1995], the *locations* of peaks in a histogram are stable under viewpoint change and scale transformation, unlike histogram bin *counts* used in [Swain and Ballard, 1991]. The storage space required is reduced when compared to using the full histogram, and standard key-based indexing techniques can be used. Even the peak locations are affected by the presence of background in the image. However, the color peaks present in an image can be determined more accurately when the histogram covers a small area of the image, minimizing the the presence of interfering colors from the background as illustrated in Figure 1.

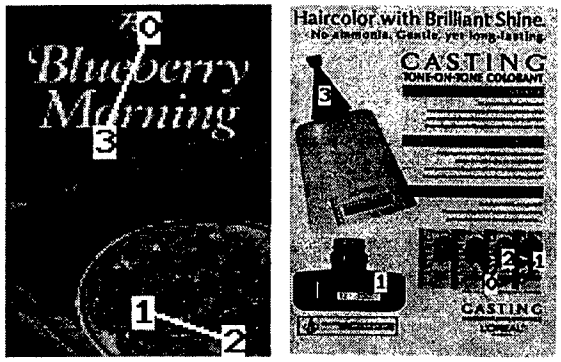
We use a *split and merge* technique for peak detection which produces accurate peaks in spite of the presence of interfering backgounds. Since

we do not know a priori the size or the location of the object of interest in the image, the image is divided uniformly into m x n cells. Local histograms are constructed for each image cell in the HSV (hue, saturation, value) color space since it is more stable than RGB under variations in illumination. Since the hue component is the most stable and value component the least stable, we use HSV histograms with finely discretized hue axis and coarsely quantized saturation and value axes (64x10x10). Peaks are detected in the histograms by finding local maxima in a 3-D neighborhood window. A combined list of peaks is produced by merging multiple copies of the same peak.

2.2 Describing spatial relationships between colors

There could be many images which have all the colors of the query object, but not in the same spatial configuration as in the query object. In the extreme case, the matched colors are scattered across the image and do not form any connected cluster. In other cases, some color adjacency relationship present in the query object may be violated in candidate images. For example, in the query in Figure 2 (a), the red (labelled 0) and blue (labelled 3) regions are adjacent whereas in the false match (b), they are not adjacent. These false matches could be eliminated if information on spatial distribution of colors in the image was available.

The color adjacency graph (CAG) formulation used by Kittler et al [Matas *et al.*, 1995] is a good descriptor of the color relationships in a multi-colored object, where the color regions are nodes in a graph with edges connecting color regions which share an edge at the pixel level. However, a CAG description of the database images is not feasible for retrieval due to the complexity of the images. Most of the images contain natural objects and color regions in which there are no distinct boundaries between colors. An attempt to construct a CAG for these images has produced very large, complex graphs, making the matching phase intractable. Therefore, we need a simpler representation for the spatial distribution of colors that allows efficient



(a) (b)

Figure 2: Example of mismatch in spatial color relationships : (a) "Blueberry Morning" query image (b) A false match

generation and storage for all images and allows fast matching.

It should be noted that during the peak detection process, we have already localized color peaks in image cells, giving us the color content in each cell. We now use this information to construct a graph describing the *approximate* spatial relationships between colors in the image *without any additional pixel level processing.*

We start by constructing an intermediate graph representation directly from the peak description of the image based on whether pixel level adjacency is *possible* between two color regions, and condense it into a compact graph - the *spatial proximity graph* (SPG). Each node in the intermediate SPG corresponds to a detected color peak, and edges between two nodes indicate that the two color regions which produced the peaks could be adjacent in the image. Let nodes of the intermediate SPG be of the form c_m^i, where m is the peak color label of the node and i is the cell in which it is located. There is an edge E between two nodes of the graph if the following condition is met.

- $E(c_m^i, c_n^j)$ if $i = j$ OR $m = n$ and (i, j) are 4-neighbors.

The intermediate graph obtained is not scale invariant, since a larger region would produce more nodes in the graph. A smaller, scale in-

variant SPG which still captures the spatial relationships between colors is obtained by *collapsing* connected nodes of the same color label into a single node of that color label. The graph may still have multiple nodes of the same color label, but only if these peaks were spatially disconnected in the image. The SPG is computed off-line for all database images and stored using an adjacency matrix representation.

The spatial proximity graph (SPG) description has a number of very useful properties. Apart from being scale and orientation invariant, it can be computed easily for all types of images, with or without prominent color boundaries. The SPG shows all possible pixel-level adjacencies that could appear in an image, without going through pixel-level processing. So any color adjacency relationship present in the image is still captured in this simplified graph. On the other hand, the graph is approximate since it may indicate some possible adjacency relationships for which there is actually no pixel-level adjacency in the image e.g. when two color regions are within a cell in the image, but there is no pixel adjacency between them.

3 Overview of Retrieval System

The features desribed above are used in an experimental retrieval system, FOCUS (Fast Object-Color-based qUery System). The schematic diagram of the system is shown in Figure 3. When a query image is obtained, the peaks in its color histogram and the graph describing the color relationships are computed.

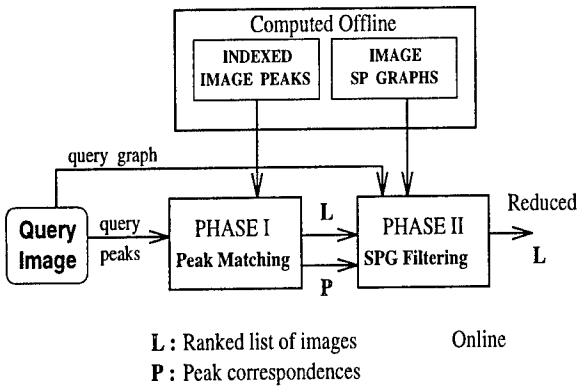
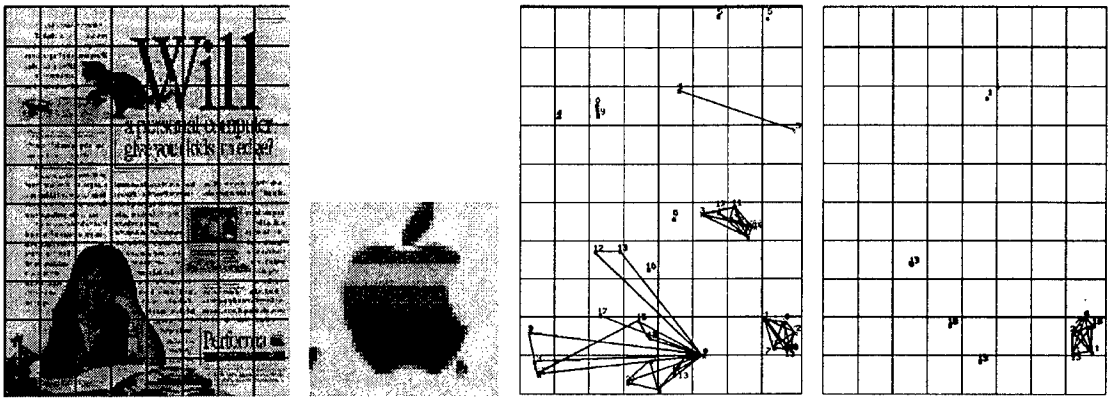


Figure 3: Overview of the FOCUS image retrieval system

The peaks extracted offline from the database images are stored in a $B+$ tree which is an order-preserving indexing structure. A *frequency table* is also constructed which gives the number of images which will be retrieved for each point in the discretized HSV space. For each peak in the query, P_q (h_q, s_q, v_q), a range query of $(h_q \pm 3, s_q \pm 4, v_q \pm 5)$ is executed starting with the peak which retrieves the minimum number of images onwards. A *join* of the lists of image identifiers is taken to find the images which have peaks matching *all* query peaks. The time complexity of the retrieval process is given by $O(qlog(kN))$, where q is the number of query peaks, N is the total number of images in the database and k is the average number of peaks per image. The images extracted are ordered by increasing mismatch scores, where the mismatch score is computed as the total *city block* distance between the matched candidate image peaks and the query peaks.

The correspondence between each color label in the image and the color peak in the query image which it matched is available from the peak matching computed during the first phase. Many image color labels may not match any query peak, since peaks maybe produced by the background in the image. The SPG computed off-line can be drastically reduced by removing all nodes in the image SPG whose color label does not match any query peak. The reduced SPG is also relabelled using the query peak color labels so that both the query graph and the reduced SPG now use the same color labels. The reduced SPG is much smaller than the original SPG, as illustrated in Figure 4, making the graph matching feasible as an online process.

The problem tackled during the online second phase is to detect if the query color graph occurs as a sub-graph of the reduced candidate image SPG. Though this is an instance of the subgraph isomorphism problem which is known to be NP-complete, due to the restricted nature of this problem, the matching computation is feasible. The running time is of the order of $O(n^m)$ where n is the size of the query adjacency matrix and m is the maximum number of instances of a color label in the reduced SPG, typically 3 or



(a) (b) (c) (d)

Figure 4: Reducing SPGs by deleting nodes not matched in phase 1: (a) Original image (b) Query image (c) SPG computed offline (d) SPG after reduction

less. Further details of this system can be found in [Das *et al.*, 1997][Das *et al.*, 1996].

4 Results

The database on which FOCUS has been tested consists of 400 advertisements from magazines and 800 color images from nature including birds, fish, flowers, animals land vegetables. The retrieval results obtained can be judged using *precision* and *recall* as criteria. Precision is the proportion of correct retrievals in the images retrieved upto the last correct image. Recall is the proportion of correct retrievals out of all the images in the database that should have been retrieved for the given On a query set of 25, the recall was 95%. The average precision after phase 1 was 44% and after phase 2 it improved to 60%. The performance was better when the query had more than three colors. The average precision score for a query set with more than three colors was 50% after phase 1 and 75% after phase 2. Two sample retrieval results are shown in Figure 5.

The time taken for a complete cycle of retrieval consists of the query processing time, phase 1 matching and phase 2 matching. FOCUS runs on a 133 MHz Pentium processor and all times mentioned are averaged over many trials. Query processing takes about 0.1 sec on a query image of size 100x200, which is the average size of queries tried. Phase 1 matching takes 0.1-0.2 sec and phase 2 matching takes about 0.01 for each image in the list produced by phase 1. Since this list has 30 images on an average the second phase takes about 0.3 sec. The retrieval process is fast enough to be scalable to very large databases since the query processing time is independent of the size of the database, the first phase of matching grows only logarithmically with the size of the database and the second phase depends only on the number of images retrieved by the first phase.

5 Conclusion

We have presented two robust color features which have been used to develop a fast, background independent color image retrieval system which produces good results with multi-colored query objects. The retrieval is robust to differences in the scale, orientation and location of the query object in candidate images. The speed of the system and the small storage overhead make it suitable for use in large databases with online user interfaces. In future, we plan to increase the size of the database, add more color features to further distinguish between images and add more phases to utilize other types of image information.

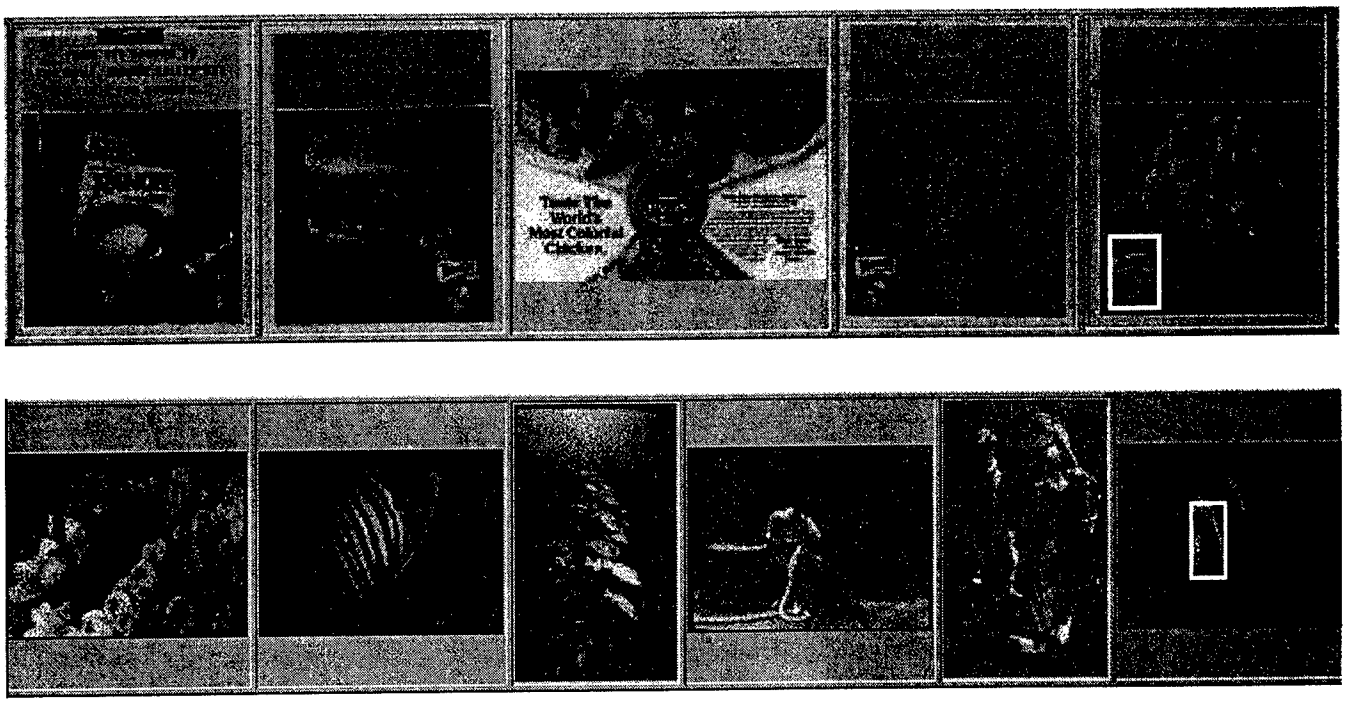


Figure 5: Examples of Retrieved Results - the query is marked by a white box

Acknowledgments

We would like to thank R.Manmatha for testing the retrieval results and providing valuable suggestions, Jonathan Lim for building the webpage for the online demonstration of the system and Bruce Draper for ideas in earlier versions of this work.

References

[Swain and Ballard, 1991] M.J. Swain and D.H. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[Niblack *et al.*, 1993] W. Niblack, R. Barber *et al.* The QBIC project: Querying Images by Content using Color, Texture and Shape. *SPIE Conference on Storage and Retrieval for Image and Video Databases*, 1908:173–187, 1993.

[Hafner *et al.*, 1995] J. Hafner, H. Sawhney, W. Niblack *et al.* Efficient Color Histogram Indexing for Quadratic Form Distance Functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729-736, 1995.

[Matas *et al.*, 1995] J. Matas, R. Marik and J. Kittler. On Representation and Matching of Multi-Coloured Objects. *Fifth International Conference on Computer Vision*, 726-732, 1995.

[Kankanhalli *et al.*, 1996] M.S. Kankanhalli, B.M. Mehtre and J.K. Wu. Cluster-Based Color Matching for Image Retrieval. *Pattern Recognition*, 29(4):701-708, 1996.

[Gong *et al.*, 1996] Y. Gong, C.H. Chuan and G. Xiaoyi. Image Indexing and Retrieval Based on Color Histograms. *Multimedia Tools and Applications*, 2(2):133-156, 1996.

[Das *et al.*, 1996] M. Das, B.A. Draper, W.J. Lim, R. Manmatha and E.M. Riseman. A Fast, Background-independent Retrieval Strategy for Color Image Databases. *Computer Science Technical report* TR-96-79, Univ. of Massachusetts at Amherst.

[Das *et al.*, 1997] M. Das, E.M. Riseman and B.A. Draper. FOCUS : Searching for Multicolored Objects in a Diverse Image Database. *To appear in IEEE conference on Computer Vision and Pattern Recognition*, 1997.

Automatic Text Detection and Recognition*

Victor Wu, R. Manmatha, Edward M. Riseman
Multimedia Indexing And Retrieval Group
(Joint Laboratory of Center For Intelligent Information Retrieval and Computer Vision Lab)
Computer Science Department
University of Massachusetts, Amherst, MA 01003-4610
E-MAIL: vwu@cs.umass.edu

Abstract

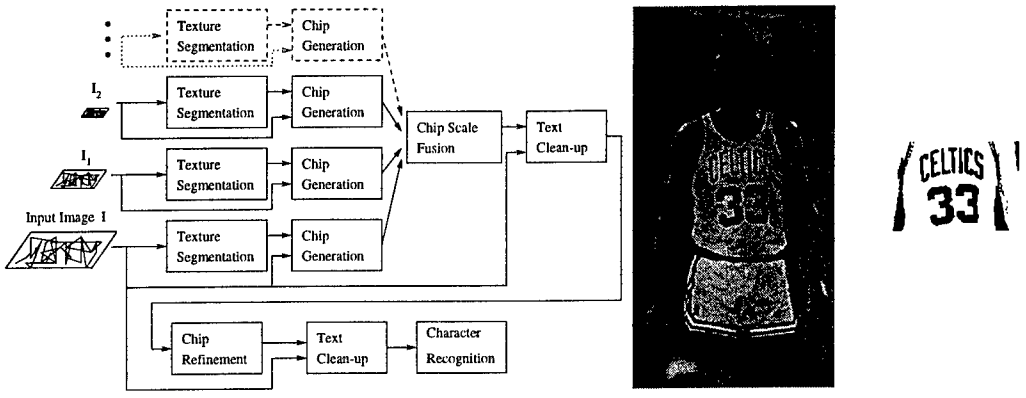
A four-step system which automatically detects and extracts text in images is presented. First, a texture segmentation scheme is used to focus attention on regions where text may occur. Second, strokes are extracted from the segmented text regions, and then processed to form rectangular boxes surrounding the corresponding text strings. Multi-scale processing is used to account for significant font size variations. Third, text is extracted by cleaning up the background and binarizing the detected text strings. Finally, better text bounding boxes are generated by using the binarized text as strokes. Text is then cleaned and binarized from these new boxes, and can then be passed through a commercial OCR engine for recognition. The system is stable, robust, and works well on images (with or without structured layouts) from a wide variety of sources, including digitized video frames, photographs, newspapers, advertisements in magazines/newspapers, stock certificates, and personal checks.

*This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, in part by the United States Patent and Trademark Office and Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235, in part by the National Science Foundation under grant number IRI-9619117 and in part by NSF Multimedia CDA-9502639. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsors.

1 Introduction

Most of the information available today is either on paper or in the form of still photographs and videos. To build digital libraries, this large volume of information needs to be digitized into images and the text converted to ASCII for storage, retrieval, and easy manipulation. For example, video sequences of events such as a basketball game can be annotated and indexed by extracting a player's number, name and the team name that appear on the player's uniform (Figure 1(b, c)). In contrast, image indexing based on image content, such as the shape of an object, is a quite difficult task.

Current OCR technology [Bokser, 1992, Mori *et al.*, 1992] is largely restricted to finding text printed against clean backgrounds, since in these cases it is easy to binarize the input images to extract text (text binarization) before character recognition begins. It cannot handle text printed against shaded or textured backgrounds, nor text embedded in pictures. More sophisticated text reading systems usually employ page segmentation schemes to identify text regions. Then an OCR module is applied only to the text regions to improve its performance. Some of these schemes [Wahl *et al.*, 1982, Wang and Srihari, 1989, Nagy *et al.*, 1992, Pavlidis and Zhou, 1992] are top-down approaches, some are bottom-up methods [Fletcher and Kasturi, 1988, O'Gorman, 1993], and others are based on texture segmentation techniques in computer vision [Jain and Bhattacharjee, 1992]. However, the top-down and bottom-up approaches usually require the input image to be binary and has a Manhattan layout. Although the ap-



(a) (b) (c)

Figure 1: The system, example input image, and extracted text. (a) The top level components of the text detection and extraction system. The pyramid of the input image is shown as I, I_1, I_2 ...; (b) An example input image; (c) Output of the system before being fed to the Character Recognition module.

proach in [Jain and Bhattacharjee, 1992] can in principle be applied to greyscale images, it was only used on binary document images, and in addition, the text binarization problem was not addressed. In summary, few working systems have been reported that can read text from document pages with both structured and non-structured layouts. The system presented in this paper is our contribution to constructing a complete automatic text reading system.

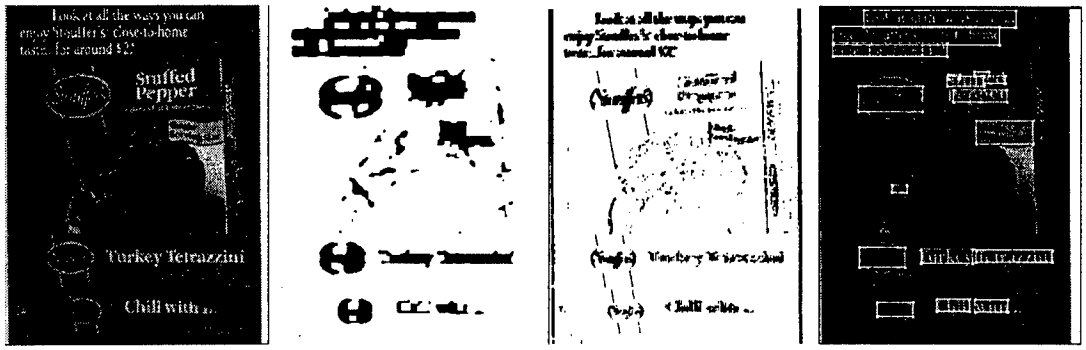
2 System Overview

Our system takes advantage of the following distinctive characteristics of text which make it stand out from other image information: (1) Text possesses a distinctive frequency and orientation attributes; (2) Text shows spatial cohesion — characters of the same text string are of similar heights, orientation and spacing.

The first characteristic suggests that text may be treated as a distinctive texture, and thus be segmented out using texture segmentation techniques. Thus, the first phase of our system is Texture Segmentation as shown in Figure 1(a). In the Chip Generation phase, strokes are extracted from the segmented text regions. Using reasonable heuristics on text strings based on the second characteristic, the extracted strokes are then processed to form tight rectangular bounding boxes around the corresponding text strings. To detect text over a wide range of font sizes, the above steps are applied to a pyramid of images generated from the input image, and then the boxes formed at each resolution level of the pyramid are fused at the original resolution. A Text Clean-up module which removes the background and binarizes the detected text is applied to extract the text from the regions enclosed by the bounding boxes. Finally, text bounding boxes are refined (re-generated) by using the extracted items as strokes. These new boxes usually bound text strings better. The Text Clean-up process is then carried out on the regions bounded by these new boxes to extract cleaner text, which can then be passed through a commercial OCR engine for recognition if the text is of an OCR-recognizable font. The phases of the system are discussed in the following sections.

3 The Texture Segmentation Module

A standard approach to texture segmentation is to first filter the image using a bank of linear filters such as Gaussian derivatives [Malik and Perona, 1990] or Gabor functions, followed by some non-linear transformation such as a hyperbolic function $tanh(\alpha t)$. Then features are computed to form a feature vector for each pixel from the filtered images. These feature vectors



(a) (b) (c) (d)

Figure 2: Results of Texture Segmentation and Chip Generation. (a) Portion of an input image; (b) The final segmented text regions; (c) Extracted strokes; (d) Text chips mapped on the input image.

are then classified to segment the textures into different classes.

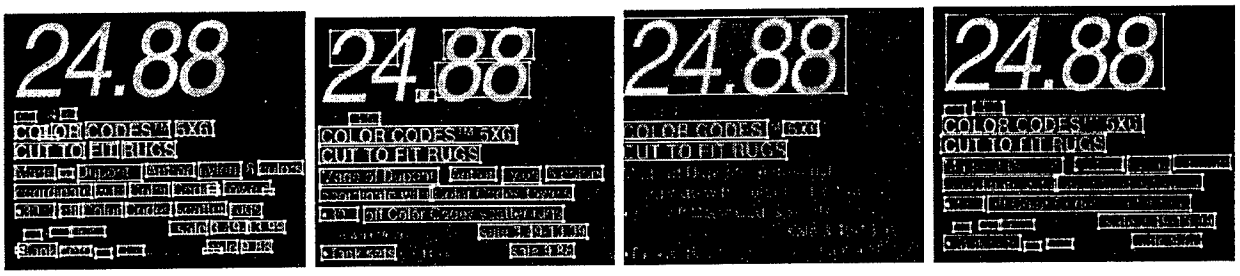
Here, 9 filters of the 3 second order derivatives of Gaussian at three different scales $\sigma = (1, \sqrt{2}, 2)$ are used. Each filter output is passed through the non-linear function $tanh(\frac{t}{4})$. At each pixel, a feature vector can be constructed consisting of the 9 energy estimates computed using the outputs of the non-linear transformation. The feature vectors are then clustered using the K-means algorithm (with K = 3). One of the clusters is labeled as text automatically. Finally, a morphological closure operation is carried out on the segmented text regions since the segmented regions might have holes and be broken [Wu *et al.*, 1997].

Figure 2(a) shows a portion of an original input image with a variety of textual information to be extracted. There is text on a clean dark background, text printed on Stouffer boxes, Stouffer's trademarks (in script), and a picture of the food. Figure 2(b) shows the final segmented text regions.

4 The Chip Generation Phase

In practice, text may occur in images with complex backgrounds and texture patterns, such as foliage, windows, grass etc. Thus, some non-text patterns may pass the filters and initially be misclassified as text (Figure 2(b)). Furthermore, segmentation accuracy at texture boundaries is a well-known and difficult problem in texture segmentation. Consequently, it is often the case that text regions are connected to other regions which do not correspond to text, or one text string might be connected to another text string of a different size or intensity. This might cause problems for later processing. For example, if two text strings with significantly different intensity levels are joined into one region, one intensity threshold might not separate both text strings from the background.

Therefore, heuristics need to be employed to refine the segmentation result. Since the segmentation process usually finds text regions while excluding most of those that are non-text, these regions can be used to direct further processing (**focus of attention**). Furthermore, since text is intended to be readable, there is usually a significant contrast between it and the background. Thus contrast can be utilized finding text. Also, it is usually the case that characters in the same word/phrase/sentence are of the same font and have similar heights and inter-character spaces. Finally, it is obvious that characters in a horizontal text string are horizontally aligned. Therefore, all the heuristics above are incorporated in the Chip Generation phase in a bottom-up fashion: significant edges form strokes (Figure 2(c)); strokes from the segmented regions are aggregated to form chips corresponding to text strings. The rectangular bounding boxes of the chips are used to indicate where the hypothesized (detected) text strings are (Figure 2(d)). These steps are described in detail in [Wu *et al.*, 1997].



(a) (b) (c) (d)

Figure 3: The scale problem and its solution. (a) Chips generated for the input image at full resolution; (b) half resolution; (c) $\frac{1}{4}$ resolution; (d) Chips generated at all three levels mapped onto the input image. Scale-redundant chips are removed.

(a) (b) (c)

Figure 4: Binarization results before and after the Chip Refinement step. (a) Magnified portion of an input image; (b) binarization result before refinement; (c) after refinement.

5 A Solution to the Scale Problem

The three frequency channels used in the segmentation process work well to cover text over a certain range of font sizes. Text from larger font sizes is either missed or fragmented. This is called the **scale problem**. Intuitively, the larger the font size of the text, the lower the frequency it possesses. Thus, when the text font size gets too large, its frequency falls outside the three channels selected in section 3.

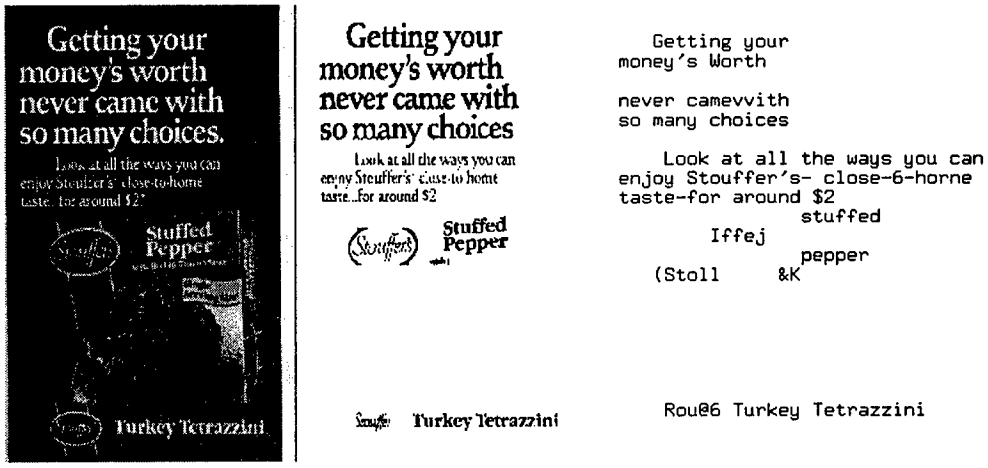
A pyramid approach (Figure 1(a)) is used to solve the scale problem: a pyramid of the input image is formed and each image in the pyramid is processed using the standard channels ($\sigma = 1, \sqrt{2}, 2$) as described in the previous sections. At the bottom of the pyramid is the original image; the image at each level (other than the bottom) has half of the resolution as that of the image one level below. Text of smaller font sizes can be detected using the images lower in the pyramid (Figure 3(a)), while text of large font sizes is found using images higher in the pyramid (Figure 3(c). The bounding boxes of detected text regions at each level are mapped back to the original input image and the redundant boxes are then removed as shown in Figure 3(d). Details are presented in [Wu *et al.*, 1997].

6 Text on Complex Backgrounds

The previous sections describe a system which detects text in images and puts boxes around detected text strings in the input image. Since text may be printed against complex image backgrounds, which current OCR systems cannot handle well, it is desirable to have the backgrounds removed first. In addition, OCR systems require that the text must be binarized

Table 1: Summary of the system's performance. 48 images were used for detection and clean-up. Out of these, 35 binarized images were used for the OCR process.

	Total Perceived	Total Detected	Total Clean-up	Total OCRable	Total OCRed
Char	21820	20788 (95%)	91%	14703	12428 (84%)
Word	4406	4139 (93%)	86%	2981	2314 (77%)



(a) (b) (c)

Figure 5: Example 1. (a) Original image (ads11); (b) Extracted text; (c) The OCR result using Caere's WordScan Plus 4.0 on b.

before actual recognition starts. In this system, the background removal and text binarization is done by applying an algorithm to the text boxes individually instead of trying to binarize the input image as a whole. This allows the process to adapt to the individual context of each text string. The details of the algorithm are in [Wu *et al.*, 1997].

7 The Text Refinement

Sometimes non-text items are identified as text as well. In addition, the bounding boxes of the chips sometimes do not tightly surround the text strings. The consequence of these problems is that non-text items may occur in the binarized image, produced by mapping the extracted items onto the original page. An example is shown in Figure 4(a,b). These non-text items are not desirable.

However, by treating the extracted items as strokes, the Chip Refinement module which is essentially similar to the chip Generation module but with stronger constraints, can be applied here to eliminate the non-text items and hence form tighter text bounding boxes. This can be achieved because (1) the clean-up procedure is able to extract most characters without attaching to nearby characters and non-text items (Figure 4(b)), and (2) most of the strokes at this stage are composed of complete or almost complete characters, as opposed to the vertical connected edges of the characters in the initial processing. Thus, it can be expected that the correct text strokes comply more consistently with the heuristics used in the early Chip Generation phase. The significant improvement is clearly shown in 4.

8 Experiments

The system has been tested over 48 images from a wide variety of sources: digitized video frames, photographs, newspapers, advertisements in magazines or sales flyers, and personal checks. Some of the images have regular page layouts, others do not. It should be pointed out

that all the system parameters remain the same throughout the entire set of test images, showing the robustness of the system.

Characters and words (as perceived by one of the authors) were counted in each image as ground truth. The total numbers over the whole test set are shown in the "Total Perceived" column in Table 1. The detected characters and words are those which are completely enclosed by the boxes produced after the Chip Scale Fusion step. The total numbers of detected characters and words over the entire test set are shown in the "Total Detected" column. Characters and words clearly readable by a person after the Chip Refinement and Text Clean-up steps (final extracted text) are also counted for each image, with the total numbers shown in the "Total Clean-up" column. The column "Total OCRable" shows the total numbers of cleaned-up characters and words that appear to be of OCR recognizable fonts in 35 of the binarized images. Note that only the text which is horizontally aligned is counted (skew angle of the text string is less than roughly 30 degrees)[1]. The "Total OCRed" column shows the numbers of characters and words from the "Total OCRable" sets correctly recognized by Caere's commercial WordScan OCR engine.

Figure 5(a) is a portion of an original input image which has no structured layout. The final binarization result is shown in (b) and the corresponding OCR output is shown in (c). Notice that most of the text is detected, and most of the text of machine-printed fonts are correctly recognized by the OCR engine. It should be pointed out that the cleaned-up output looks fine to a person in the places where the OCR errors occurred.

9 Conclusion

A robust system has been presented which automatically detects and extracts text from images from a wide variety of sources such as newspapers, magazines, printed advertisement, photographs, and checks. The application potential of the system is enormous.

[1] In this paper, the focus is on finding horizontal, linear text strings only. The issue of finding text strings of any orientation will be addressed in future work.

References

[Bokser, 1992] Mindy Bokser. Omnidocument Technoligies. *Proceedings of The IEEE*, 80(7):1066–1078, July 1992.

[Fletcher and Kasturi, 1988] Lloyd Alan Fletcher and Rangachar Kasturi. A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 10(6):910–918, Nov. 1988.

[Jain and Bhattacharjee, 1992] Anil K. Jain and Sushil Bhattacharjee. Text Segmentation Using Gabor Filters for Automatic Document Processing. *Machine Vision and Applications*, 5, 1992.

[Malik and Perona, 1990] Jitendra Malik and Pietro Perona. Preattentive texture discrimination with early vision mechanisms. *J. Opt. Soc. Am.*, 7(5):923–932, May 1990.

[Mori *et al.*, 1992] S. Mori, C. Y. Suen, and K. Yamamoto. Historical Review of OCR Research and Development. *Proceedings of The IEEE*, 80(7):1029–1058, July 1992.

[Nagy *et al.*, 1992] G. Nagy, S. Seth, and M. Viswanathan. A Prototype Document Image Analysis System for Technical Journals. *Computer*, pages 10–22, July 1992.

[O'Gorman, 1993] Lawrence O'Gorman. The Document Spectrum for Page Layout Analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(11):1162–1173, Nov. 1993.

[Pavlidis and Zhou, 1992] Theo Pavlidis and Jiangying Zhou. Page Segmentation and Classification. *CVGIP: Graphical Models and Image Processing*, 54(6):484–496, Nov. 1992.

[Wahl *et al.*, 1982] F. M. Wahl, K. Y. Wong, and R. G. Casey. Block Segmentation and Text Extraction in Mixed Text/Image Documents. *Computer Graphics and Image Processing*, 20:375–390, 1982.

[Wang and Srihari, 1989] D. Wang and S. N. Srihari. Classification of Newspaper Image Blocks Using Texture Analysis. *Computer Vision, Graphics and Image Processing*, 47:327–352, 1989.

[Wu *et al.*, 1997] Victor Wu, R. Manmatha, and Edward. M. Riseman. Finding Text In Images. *Technicial Report 97-09, Computer Science Department, UMass, Amherst, MA*, 1997.

SECTION III
IMAGE UNDERSTANDING ENVIRONMENT
(IUE)

IMAGE UNDERSTANDING ENVIRONMENT (IUE) TECHNICAL PAPERS

The Image Understanding Environment Progress since IUW'96*

Richard A. Lerner
Amerinex Applied Imaging, Inc.
409 Main St., Amherst, MA 01002
E-MAIL: rlerner@aai.com
http://www.aai.com/

Abstract

The Image Understanding Environment (IUE) provides a large, robust, well-documented, C++ hierarchy and tool suite to support Image Understanding research and technology transfer. This paper describes the additions and improvements that have been made to the IUE since its first public release at IUW'96.

1 Introduction

The Image Understanding Environment (IUE) is a software environment intended for use in developing image understanding algorithms and applications. Sponsored by DARPA, it was conceived to promote the exchange of research results within the IU community by providing a common environment that can be used and extended throughout the community. By enabling this exchange, the IUE enhances the productivity of the community and provides the means to quantitatively measure progress in the field.

More specifically, the IUE provides a well-documented, modular, object-oriented, C++ class hierarchy; a suite of development tools; implementations of established IU algorithms; and the ability to interoperate with existing environments. Papers in previous IUW Proceedings provide more detailed descriptions of the IUE components. Another paper in this year's proceedings demonstrates many capabilities of the IUE by looking at the implementation of a real application. This paper describes the status of the IUE and our plans for its future.

The IUE project is currently in the fourth year of a five year development program. The first public release of the IUE was announced at last year's Image Understanding Workshop. That release demonstrated the core components of the IUE hierarchy and tools, and included a few demonstration algorithms implemented using the IUE. Since that time, the IUE team has focused on improving the performance of the system, enhancing its tools, and creating a core library of IU algorithms. The remainder of this paper describes these improvements, describes our plans for the future, and provides information on how to obtain the current version of the IUE via anonymous ftp.

2 Performance and Ease of Use Improvements

2.1 Size and Speed

The IUE class hierarchy is a very large. It currently contains specifications for approximately 750 classes and implementations for 600. Including template instances, the IUE libraries contain close to 1000 classes. With a system this large, careful planning is needed to ensure that the libraries can be effectively used. Over the last year, the IUE team focused on reducing the size of the IUE libraries to manageable levels. Table 1 shows some approximate statistics for version 1.1, released last year, and for the current version (as of this writing, v2.0-beta).

*This work supported by ARPA under TEC contract DACA76-93-C-0015.

Metric	v1.1	v2.0-beta
Number of classes	460	600
Number of template instances	430	400
Size of libraries (Sun4/Solaris)	122MB/130MB	24MB/26MB
Typical compilation time	25 sec (Sun4)	10 sec (Solaris)
Typical link time (w/dynamic linking)	100 sec	25 sec
Typical load time (dynamic linking)	50 sec	5 sec

Table 1: Size and speed improvements.

The most significant improvements came from reducing the size of the libraries from 122MB to 24MB. These smaller libraries result in significantly faster compile/link/run cycles. We obtained this reduction, despite adding 100 new classes, with improvements such as the following:

- Better encapsulation of Standard Template Library (STL) and other templates, so they would not be instantiate in every module
- decoupling modules to reduce the number of symbols duplicated in multiple modules
- reducing the amount of code generated for each IUE class by our code generator
- shortening names used as template parameters (STL template symbol names can grow to more than 1K)
- grouping related template instances into single modules
- removing unnecessary intermediate classes from our hierarchy

2.2 Installation

Early users of the IUE often experienced difficulties installing the IUE. Most of the problems involved mismatches between the user's environment and the environment the IUE expected to find. These differences included tool version mismatches, improper configuration, or inadequate forethought on the part of the IUE developers. To combat these problems, we consolidated all user provided configuration information into a single file, provided a tool to validate the user's environment, upgraded the IUE to use the latest versions of the GNU compiler and libraries, and, where possible, made the system less dependent on particular configurations. We also improved the installation document and provide instructions on validating the IUE installation, once it is complete.

2.3 Documentation

From the outset, everyone involved with the IUE recognized the importance of good documentation. For this reason, the interfaces to all of the IUE classes are generated from specifications that are also used to generate documentation, both printed and electronic (HTML). Generating code from a specification in this way ensures that the reference manual is always up-to-date. It also encourages implementors to provide documentation as part of the design/implement process (since they need the spec to generate the code). In addition, it enables the use of a variety of tools, such as interface definitions for different languages and documentation formats, to generate useful information from the specification.

The IUE includes five major documentation components:

- Primer—detailed, step-by-step instructions and illustrations on how to use the IUE to solve common problems.
- Overview—high-level discussion of the IUE programming model and class structure, with discussions of the major sub-trees.
- Programmers Reference—detailed information about every class in the IUE.

- Data Exchange Reference—complete definition of our exchange files and descriptions of the components available to assist in reading and writing files from non-IUE systems.
- Installation guide and Release notes—specific instructions and information about the release.

Since the first public release, we have significantly improved the Programmers Reference and Installation guide, incorporating many ideas from our early users, and have added and updated a number of Chapters in the Primer. The most significant additions to the Primer over the past year include:

- a new chapter on using the IUE to implement grouping algorithms,
- a new chapter describing the IUE visualization tools, and
- an updated chapter describing the creation and use of IUE tasks.

3 Development Tools

3.1 Visualization

The IUE's development tools, particularly the visualization tools, have undergone significant revision since the first public release. The Alpha visualization tool released last year has been replaced by a new program built on top of Fresco (a public-domain successor to Interviews). This new version can display more data types, support multiple views, additional interaction, and provides a cleaner interface. We have also provided a simple functional interface that allows users to write programs that dynamically add data to a display. A new demonstration program, `iue-examples`, demonstrates the use of this interface, as well as demonstrating many of the tasks in the IUE task library.

The newest addition to the IUE's tool suite is a Java version of the visualization program. This program, written entirely in Java, includes many of the capabilities of the Fresco based visualization tool. In its current form, the Java interface reads data from IUE Data Exchange files and allows users to display the data and manipulate the data. In the near future, we expect this tool to be able to communicate with an IUE server that allows users to invoke IUE tasks on data selected within the display. Ultimately this capability should allow users to build applications using Java as a sophisticated scripting language. The functionality of the Java tool is still limited; the performance of the Java visualization tool is presently not as good as that of the Fresco tool, especially when reading DEX files, and it can only display 2D data. However, we will eventually add 3D visualization and we expect performance to significantly improve over time, especially as new compilers become available.

3.2 Khoros compatibility

Over the last year the IUE team has continued to improve compatibility between the IUE and Khoros. The IUE and Khoros do not depend on each other in terms of installation, code dependencies, or core functionality. However, using them together can provide substantial benefits—IUE users can use existing libraries of image processing algorithms written for Khoros, and Khoros users can use the IUE to perform feature-based processing, either using existing IUE tasks, or writing their own using the IUE representations.

In its current form, the IUE—Khoros link includes:

- A tool to create Khoros glyphs to represent IUE tasks on Khoros' Cantata desktop
- Pre-made glyphs for all tasks in the IUE Algorithm Libraries Support, in the IUE, for reading and writing KDF image files to allow IUE glyphs to pass image data to and from Khoros glyphs
- Wrappers that allow IUE glyphs to communicate symbolic data using IUE's DEX files.

The IUE currently supports the most recent version of Khoros (version 2.1 at the time this of

writing).

4 Task Libraries

The IUE was designed with the model that, once the infrastructure was in place, the task libraries would become populated with user-contributed algorithms. This exchange of software holds the greatest potential value of the IUE. However, a core library is necessary to validate the IUE implementation and design, and to bootstrap development by encouraging users to begin using the IUE.

The first public release included example applications that demonstrated the use of the IUE for a few tasks. These included a collection of tasks to detect features in solar imagery, a SAR point detector, a FLIR segmenter, and a collection of IUE and KBVision tasks that perform model-based change detection.

Since the version 1.1 release last year, the IUE team has been working on populating the "official" IUE Task Library with robust implementations of a collection of common and/or current algorithms. Tables 2 and 3 lists the contents of the task library as of version 2.0-beta. In the selection of algorithms for this library, we focused on tasks that create and manipulate features, since these algorithms highlight some of the unique capabilities of the IUE, and they are less widely available than standard image processing algorithms. In the near future, we expect to add a collection of image processing tasks to the IUE libraries. Until that time, users can use the algorithms available in Khoros, or any other image processing algorithm that can produce image files in a format understood by the IUE (i.e., Tiff, KBVision, or KDF file format).

As the IUE implementors, Amerinex supports three mechanisms for distributing algorithm implementations among IUE users. Algorithms that are sufficiently robust and properly documented can be added to the "official" IUE task libraries, where they will be maintained as the IUE evolves. Algorithms that are too complex to be maintained, or do not meet the criteria for inclusion in the "official" library, can be added to "user contributed" libraries that are distributed along with the IUE. Finally, the IUE will publish references to IUE task libraries that are maintained (or not) at the author's site. The "user contributed" libraries currently contain the tasks in Table 4, which are actually IUE wrappers around non-IUE code provided by the contributors, and the tasks that make up the demonstration applications. Over time we expect to "harden" the demonstration tasks and migrate them to the "official" library.

5 Class Hierarchy Additions

Over the last year most of the class implementation effort has focused on implementing the basic 3D representations. In particular, the IUE now includes various 2D and 3D curves, 3D planes and patches, and soon, 3D functional surfaces. Other additions include:

- support for Khoros' KDF image file format
- image-feature-collection
- circular arcs
- enhanced graph classes
- tuples and intervals
- basic spatial index classes
- 1D and 2D histograms
- pixel chains
- reorganized coordinate system and transform classes and a new coordinate transform graph interface.

6 User community

Our user community has been steadily growing since the IUE's first public release. Some of the more substantial efforts include the following:

UK initiatives: The Image Understanding Community in the UK has received funding from their government to evaluate and extend the IUE. Eight research sites are working with the IUE to determine how their environments can be integrated with the IUE.

Function	Name	Basic Description
Point detection	extremal-points	Locates local extrema of the image intensity function within a specified window
	extremal-curvature	Locates points in the image intensity function where the magnitudes of the principal curvatures are both large
	Deriche-Giraudon-corners	Implements the corner detection process described by Deriche and Giraudon [1991]
	Kitchen-Rosenfeld-corners	Implements the corner detection process described by Kitchen and Rosenfeld [1982]
Edge detection	Marr-Hildreth-edgels	Implements the edge detection process defined by Marr and Hildreth [1980]
	Canny-edgels	Implements the detection process described in [Canny, 1986]
	valley-ridges	Locates points in image intensity function where the magnitude of the largest principal curvature is great and the magnitude of the smallest principal curvature is near zero
	Frei-Chen-edgels	Implements the Frei-Chen boundary detection algorithm as described in [Frei and Chen, 1977]
Fitting	line-fitting	Implements the direct algebraic technique for fitting a line to a set of points as described in [Agin, 1981]
	plane-fitting	Implements the direct technique for fitting a surface to a set of points as described in [Pratt, 1987]
	ellipse-fitting	Implements an efficient and robust method for fitting ellipses to scattered data as described by Fitzgibbon, Pilu, and Fisher [1996]
	circle-fitting	Implements the method for directly fitting a general implicit quadratic surface to a set of 2d points as described in [Agin, 1981], but constrained to fit a circle
	conic-fitting	Implements the method for directly fitting a general implicit quadratic surface to a set of 2d points as described in [Agin, 1981]
	cubic-fitting	Implements the direct technique for fitting a surface to a set of points as described in [Pratt, 1987]

Table 2: IUE task library functions.

Function	Name	Basic Description
Curve formation	edge-contours	Implements simple grouping algorithm that chains edgels together to form edge contours
	Glazer-chains	Implements the grouping process described in [Glazer, 1992]
	Guy-Medioni-curves	Computes a set of "saliency maps" as described by Guy and Medioni [1993]
	Pavlidis-Horowitz-polylines	Implements the algorithm, described by Pavlidis and Horowitz [1974], for approximating a sequence of points by a set of lines
	Pavlidis-polylines	Implements a variant of the "Polygonal Fit" algorithm described in [Pavlidis, 1982]
	Sobel-edges	Implements a relatively simple method to recover extended edge structures, represented as pixel-chains
Region formation	adaptive-region-growing	Implements an algorithms similar to simple-region-growing, except that pixel differences are compared against an adaptive threshold
	simple-region-growing	Implements a simple queue-based method for region growing

Table 3: IUE task library functions (continued).

The University of Manchester, is coordinating these efforts and is acting as a central repository to maintain and distribute the results of their efforts. They are also acting as the primary line of communication between their efforts and our development work. David Cooper has principal responsibility for this project at the University of Manchester.

Target Jr.: Joe Mundy's laboratory at General Electric has begun the process of integrating GE's TargetJr environment with the IUE. They are developing the C++ visualization infrastructure that both the IUE and TargetJr use, and are developing the necessary linkage between TargetJr and IUE classes. The ultimate goal is for the IUE to subsume the capabilities of TargetJr.

Terry Boult: Terry Boult's laboratory at Lehigh University has embarked on a number of projects to enhance current IUE capabilities. These projects include the design and implementation of a filter class hierarchy, interactive interfaces to the IUE, improved matrix classes, and sensor and sensor models. Terry Boult is also primarily responsible for the IUE code generator.

IUE Summercamps: Terry Boult expanded the IUE summercamp program, organizing two summercamp sessions last summer. The first took place at Lehigh. Approximately 10 IU graduate students attended from IU programs at various universities. The second session took place in Europe, for the benefit of European IU community. This session was attended by approximately 12 students and researchers from throughout europe.

SGI port: Lee Iverson at SRI is working on porting the IUE to IRIX. We expect to incorporate his modification and distribute the IRIX configuration files in the relatively near future. AAI will not provide full support for this port, but will act as a clearing house for changes and updates.

Function	Name	Basic Description
Stanford	Wang-Binford-chains	IUE wrapper around code provided by authors
Brunel	Rosin-West-arcs	IUE wrapper around code provided by authors
CMU	point-detection	IUE wrapper around code provided by K. Ikeuchi to detect peaks in SAR images
AAI	gaussian-smooth	Smooths a scalar image using scaled-integer approximation to Gaussian kernel
	map-lines-to-image	Maps a set of lines to the equivalent raster, where each line's raster has a user-specified width
	quantization	Maps a scalar image through uniform quantization
	connected-components	Maps a label image to an image where each connected region has a unique new label
	planar-correction	Fits a plane to the intensity data of a scalar image and remaps the image pixels using the plane to normalize values
	AAI-valley-detection	Computes valley edgels by analyzing the principal curvatures of the intensity surface

Table 4: User-contributed task library functions.

7 Plans for the Future

Development efforts for the remainder of year four will focus on the following areas:

- continuing efforts to make the IUE more "usable" according to metrics such as compile/link/run cycle time
- continuing to improve and expand the capabilities of the visualization tools
- reworking the image class implementation to better support large images and to improve image access performance in general
- adding 3D algorithms to the task library
- providing support to our growing user community

Terry Boult is again organizing a summercamp session at Lehigh University this year. Anyone interested in attending should send mail to *iue-help@aai.com.*

8 Obtaining the IUE

The IUE version 2.0 consists of the IUE class library, including complete specification and sources, HTML and PostScript documentation, a primer, and support libraries. In addition, pre-compiled libraries are available for our supported architectures: SunOS4, Solaris, and Linux2. The IUE is available via anonymous FTP from Amerinex and a number of mirror sites in the US, Canada, Europe, and Japan. To get full information on ftp and web access to the IUE, send email to *iue-info@aai.com* with the subject "HELP", or visit Amerinex's web site at http://www.aai.com. To join the iue-users mailing list, send email to *iue-users-request@aai.com*

References

[Agin, 1981] G. Agin. Fitting ellipses and general second-order curves. Robotics institute, Carnegie-Mellon University, Pittsburgh, PA, 1981.

[Canny, 1986] John Canny. A computational approach to edge detection. *IEEE PAMI*, 8(6):679–698, 1986.

[Deriche and Giraudon, 1991] R. Deriche and G. Giraudon. Accurate corner detection: An analytical study. Robotics, Image, and Vision Program 1420, INRIA, France, April 1991.

[Fitzgibbon *et al.*, 1996] A. Fitzgibbon, M. Pilu, and R. Fisher. Direct least squares fitting of ellipses. Department of artificial intelligence, University of Edinburgh, Edinburgh, Scotland, 1996.

[Frei and Chen, 1977] W. Frei and C. Chen. Fast boundary detection: A gernalization and a new algorithm. *IEEE Transactions on Computers*, 26(10):988–998, 1977.

[Glazer, 1992] F. Glazer. Fiber identification in microscopy by ridge detection and grouping. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pages 205–212, Palm Springs, CA, 1992.

[Guy and Medioni, 1993] G. Guy and G. Medioni. Inferring global perceptual contours from local features. In *Proc. DARPA IUW*, pages 881–892, 1993.

[Kitchen and Rosenfeld, 1982] L. Kitchen and A. Rosenfeld. Gray-level corner detection. *Pattern Recognition Letters*, pages 95–102, December 1982.

[Marr and Hildreth, 1980] D. Marr and E. Hildreth. Theory of edge detection. In *Proc. R. Soc. London*, pages 187–217, 1980.

[Pavlidis and Horowitz, 1974] T. Pavlidis and S. Horowitz. Segmentation of plane curves. *IEEE Transactions on Computers*, pages 859–870, August 1974.

[Pavlidis, 1982] T. Pavlidis. *Algorithms for Graphics and Image Processing.* Computer Science Press, Rockville, MD, 1982.

[Pratt, 1987] V. Pratt. Direct least-squares fitting of algebraic surfaces. *Computer Graphics*, 21(4):145–152, 1987.

Programming in the Image Understanding Environment: Locating Fibers in Microscope Images*

Richard A. Lerner
Amerinex Applied Imaging, Inc.
409 Main St., Amherst, MA 01002
E-MAIL: rlerner@aai.com
http://www.aai.com/

John Dolan
Amerinex Applied Imaging, Inc.
409 Main St., Amherst, MA 01002
E-MAIL: jdolan@aai.com

Abstract

The IUE provides a large, robust, well-documented, C++ hierarchy and tool suite to support Image Understanding research and technology transfer. This paper explores many of the capabilities of the IUE by looking closely at the implementation of an application to locate fibers in a microscope image. Our example explores concepts such as image feature representations, spatial indices, graphs, matrices, points and curves. We also discuss the IUE visualization library and tools, and demonstrate some of their capabilities.

1 Introduction

The Image Understanding Environment (IUE) is a software environment for developing image understanding algorithms and applications. Sponsored by DARPA, it was conceived to promote the exchange of research results within the IU community by providing a common environment that can be used and extended throughout the community. By enabling this exchange, the IUE enhances the productivity of the community and provides the means to quantitatively measure progress in the field.

More specifically, the IUE provides a well-documented, modular, object-oriented, C++ class hierarchy; a suite of development tools; implementations of established IU algorithms; and the ability to interoperate with existing environments. Papers in previous IUW proceedings [Dolan *et al.*, 1996, Kohl *et al.*, 1994] provide more detailed descriptions of the IUE components. Another paper in this year's proceedings describes the current status of the IUE and our plans for its future. This paper illustrates some of the capabilities of the environment by exploring a complete application written using the IUE class hierarchy, task libraries, and user interface. We begin with a description of the problem to be solved. The following sections describe various solutions using the IUE. We end by discussing the IUE's application model and visualization tools.

2 Problem: Fiber Extraction

The fiber extraction application locates the position, shape and extent of fibers in microscope imagery. It is intended to be part of a quality control system that evaluates a manufacturing process by measuring characteristics of fibers—such as size, density, count, and length—as viewed in a microscope image. The extraction process scans an image and generates a collection of curves representing the fiber segments it finds. Later processing uses these curves to compute the desired quality metrics.

The fiber extraction process follows a typical formula:

1. image processing
2. feature extraction
3. feature grouping
4. analysis

*This work supported by ARPA under TEC contract DACA76-93-C-0015.

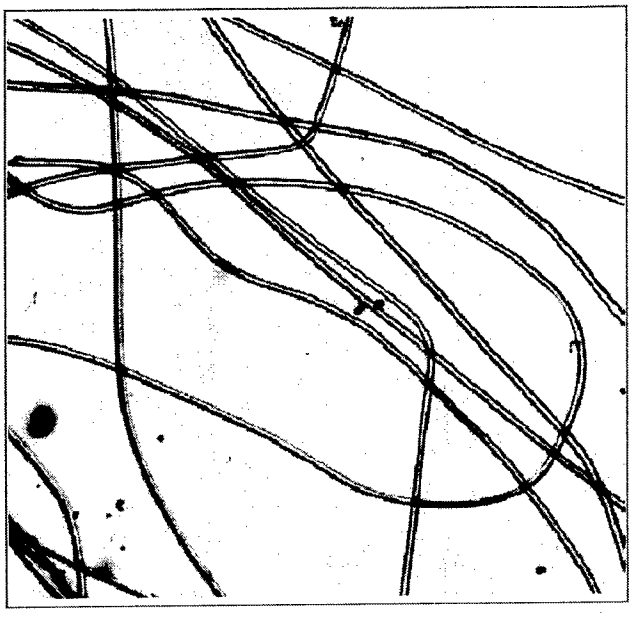


Figure 1: A typical fiber image.

The images we use in this example, are white-light images with 512 × 474 scalar pixels. Depending on fiber properties, lighting, and other optical effects, fibers may appear as two parallel curves (valleys in an intensity image) with a small ridge between them, as shown in Figure 1. The image processing stage cleans up the image and removes the ridge between the valleys. The feature extraction stage locates points within the image that show the characteristics of a valley. The feature grouping stage forms the valley points into curves. The analysis stage performs the application-specific analysis on the curves such as computing the quality metrics. We discuss the first three stages in sections 3, 4, and 5. The fourth stage is beyond the scope of this paper. Section 6 demonstrates the tools for combining the stages (as IUE tasks) into a single program that computes the curves from an image and displays the partial and final results.

3 Preprocessing

In typical computer vision applications, preprocessing usually involves the application of image processing techniques to an original source image. The operations commonly used include smoothing, thresholding, edge sharpening, histogram equalization, background subtraction, and image quantization. In each case, the objective is to enhance particular properties or correct certain deficiencies in the image so that subsequent visual processing is simplified. The particular operation may be defined over a single pixel, or a neighborhood of pixels, and it may be applied to the entire image, or restricted to a region of interest.

The IUE, like most environments for Image Understanding research, provides libraries of basic image processing tasks. The current libraries include tasks that perform Gaussian-smoothing, thresholding, quantization, planar correction, and histogram computation. Soon we will add a collection of filter classes and additional tasks that use these filters to perform image processing functions more efficiently.

3.1 Problem: Reducing Noise

The first processing stage in the fiber extraction application entails processing the image to enhance the characteristic evidence of fibers in an image, while reducing unimportant information. In our example, later processing stages will look for valleys that are characteristic of fibers in an image. To improve the performance of these later stages, we smooth the image by applying a Gaussian filter. This operation makes the valleys more pronounced by removing noise along the edges, and removes uninteresting details that might lead to the detection of spurious valleys. This operation is valid (it does not introduce or discard relevant information) since true fiber valleys are pronounced and relatively large.

We use the IUE task `AAI_gaussian_smooth` to smooth the image, as shown in Example 1. This code simply invokes the task with an image it reads from a file, to obtain a new, filtered, image.

The complete fiber application then follows this task by invocations of the tasks described in the next two sections. Section 6 shows the code that performs these operations, along with calls to the visualization interface to display partial and final results.

```
IUE_image_pointer im_ptr_in, im_ptr_out;
IUE_INT order_x, order_y;

// Load the image from a file
IUE_scalar_image_2d* im_in = read_image( input_image );

// Create an image pointer object
im_ptr_in.put_image_ptr(im_in);

// Invoke the AAI_gaussian_smooth task
AAI_gaussian_smooth(im_ptr_in, order_x, order_y, im_ptr_out);
```

Example 1: Invoking the Gaussian Filter Task.

4 Image Feature Extraction

The early stages of visual processing are typically concerned with directly computing information from the image in the form of simple primitive features like points, edges, lines, textures, and regions. Processing is generally data-driven and bottom-up, and the computation is usually local and identical over all image positions. The result may be a new image or set of images (e.g., the *intrinsic images* of Barrow and Tenenbaum [1978]); or it may be a set of *image features*, which are geometric, symbolic entities encapsulating structural properties of the image.

Because image features are geometric in nature, these classes inherit from spatial objects—indeed, each instantiable image feature *is* a spatial object. And because they originate in the image, it is desirable to provide additional attributes that connect image features to their source image and that characterize the signal properties of that image. To this end, the image feature hierarchy is defined as a set of *mixin* classes, that combine with the spatial object hierarchy to encapsulate relevant image properties for their instantiable descendants. The class IUE_image_feature provides a common root for this mixin hierarchy.

Another significant aspect of image features is that the processes used to compute them often produce a collection of feature objects rather than a single object. For example, edge detection typically produces a set of edgels corresponding to significant edge locations in the image (as opposed to a single isolated edge location). For later visual tasks, it is frequently desirable to identically process each member of this collection. In addition, because all members of the collection are derived from the same image, they share a common coordinate system. Thus, it may be useful to define a *spatial index* on the collection to make subsequent processing more efficient. The container class IUE_image_feature_collection provides such expedients.

4.1 Representing image events

Although it is possible to represent the computed image events using a so-called *iconic* scheme like intrinsic images, such a representation has a number of drawbacks in contrast to a symbolic scheme like image features. The primary differences between the two representational strategies include the following factors:

Resolution — Iconic representations have a fixed and uniform resolution—usually the pixel sample rate of the source image. Symbolic representations support sub-pixel and non-uniform resolutions, which means that they can represent objects in $\mathcal{R}^n$.

Correspondence — Iconic representations are dense in the sense that they have a value at every image location, whereas image events are typically sparse. For example, it would be quite unusual to have an edge at each image location. Since

symbolic features are created in one-to-one correspondence with image events rather than with image locations they are thus sparse like the events themselves. Furthermore, iconic representations support only one value per image location. Although multi-valued images (e.g., tuple-images) and multiple images could be used, it is somewhat problematic to represent more than one event at a location. Yet this may often be a requirement, e.g., where two edges intersect or where a curve and a region abut.

Representational character — Iconic representations are implicit; symbolic representations are explicit. For example, edges extracted from a source image might be represented as a field of edge magnitudes in one intrinsic image and edge directions in another. Note that at any location there is no explicit connection between the respective values in the two images. More precisely, in the iconic representation there is no first-class object representing the edge; instead, it must be recovered or inferred from the individual properties recorded at a location.

Geometry — The geometry of iconic representations tends to be spatially distributed. Take for example the problem of representing a curve like an ellipse. One could simply mark in a single image the locations that the curve passes through; thus, the representation of the curve is distributed over the pixels it intersects. However, such a scheme leaves important properties of the curve implicit—e.g., given a point on the curve, what is the tangent there? What is the next point? Where is the center? In contrast, image features are coherent geometric entities. One has only to create an ellipse object from the particular parameters and the curve is represented coherently as a single integral unit.

Functional attachment — As images, iconic representations have little functional attachment that is useful for subsequent computations. Whereas image feature classes provide methods that naturally support grouping and structural inference such as the geometric methods derived from spatial objects, images mainly provide access and iteration constructs alone.

The purpose of the foregoing discussion is not so much to disparage iconic representations as to point out the representational and computational advantages of image feature classes. Based on these comparisons, it should be apparent that iconic schemes can reasonably support the representation of only the most primitive initial features extracted from the image. Anything beyond such simple structures as points, edges, and the like, requires a more geometric, symbolic approach. Since the IUE provides a rich set of image features that are carefully matched to particular image events, the use of image feature classes is thus preferred in the ensuing discussions and in general.

4.2 Valley Points

When one considers the image intensity function as a surface, there are a number of significant categories of point events that indicate the topographical nature of the surface in a local neighborhood. These include *peaks*, *pits*, *ridges*, and *valleys*. Each type of point is distinguished by its principal curvatures, $\kappa_{min}, \kappa_{max}$, as follows.

peak Both principal curvatures are negative: $\kappa_{min}, \kappa_{max} < 0$.

pit Both principal curvatures are positive: $\kappa_{min}, \kappa_{max} > 0$.

ridge The magnitude of the maximum curvature is small relative to that of the minimum curvature, which is negative: $|\kappa_{max}| \ll |\kappa_{min}|$ and $\kappa_{min} < 0$.

valley The magnitude of the minimum curvature is small relative to that of the maximum curvature, which is positive: $|\kappa_{min}| \ll |\kappa_{max}|$ and $\kappa_{max} > 0$.

Apart from differential geometry, what lends significance to each event (i.e., what makes it

worth computing) is its connection to the specific physical phenomena being imaged. For the problem at hand, the key correspondence is that which exists between valley points of the image intensity function and the presence of *fibers*.

The IUE provides image point classes (e.g., IUE_image_point_2d) to represent peak and pit events, and point edgel classes (e.g., IUE_point_edgel_2d) to represent ridge and valley events. The code shown in the following examples is extracted from the task `AAI_valley_detection`. This task is specialized for extracting IUE_image_point_2d events corresponding to valleys in an image. The IUE also provides a more general task, `valley-ridges`, that extracts either point or segment edgels that correspond to valley and/or ridge edgels. Another IUE task, `Extremal-curvature`, extracts image-points that correspond to peaks and/or pits.

4.3 Valley point detection

The second fiber processing phase involves locating valley points that we can later group along the valley floor into curvilinear structures. Because of this subsequent grouping step, we use edgel features, specifically IUE_point_edgel_2d, to record point estimates, rather than simple point features such as IUE_image_point_2d). The edgel class supports the representation of direction as well as location information—both of which are used in the curvilinear grouping operation. The next section looks at the problem of grouping edgels into curve fragments. Here the focus is on computing initial local estimates of valley points.

Finding valley points entails finding points on the image surface where the maximum curvature is relatively large and the minimum curvature is small—i.e., points where the surface is relatively flat along the feature, but rapidly curves upward in the transverse direction—cf. Figure 2(b). More than that, the points should lie on the valley floor, which means that they should be locally minimum in the direction of maximum curvature. The two main steps in the detection process are therefore: 1) compute the principal curvatures to determine if the topography is valley-like; 2) interpolate the surface in the direction of maximum curvature to find the locally minimum point.

To actually compute the curvatures at a point in the image, one could explicitly convert the image to an IUE_discrete_functional_surface, S, using the appropriate constructor on that surface class. Then at each surface location $S(x, y)$, the curvatures are simply found using the methods minimum-curvature and maximum-curvature. Likewise, the direction of κ_{min} is easily obtained by invoking the method principal-axis-min.

Alternatively, and as is shown in the code below, one could compute the partial second derivatives of the image function I via convolution and at each image location form the Hessian H from these partials. The Hessian of I is defined as the following 2×2 matrix.

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix}$$

where the elements are the second partial derivatives of I at location (x, y). It can be shown that the eigenvalues of this system are proportional to the principal curvatures of the surface at (x, y) and that the corresponding eigenvectors give the principal directions. In fact, provided that the surface is oriented with *normals* pointing in the direction of positive z, then for each (x, y) of the image the direction of the largest directional second derivative is given by the eigenvector associated with the largest eigenvalue of H, which in turn corresponds to the direction of maximum surface curvature.

The code fragment in Example 2 displays part of the computation at each pixel. In particular, it shows initialization of the Hessian and the computation of the eigenvector associated with the largest eigenvalue—equivalently the direction of maximum surface curvature. Note that the `threshold_in` parameter ensures that only pronounced valleys (as opposed to gradual depressions) are considered.

This code makes use of the IUE matrix and vector classes. The IUE supports a full complement

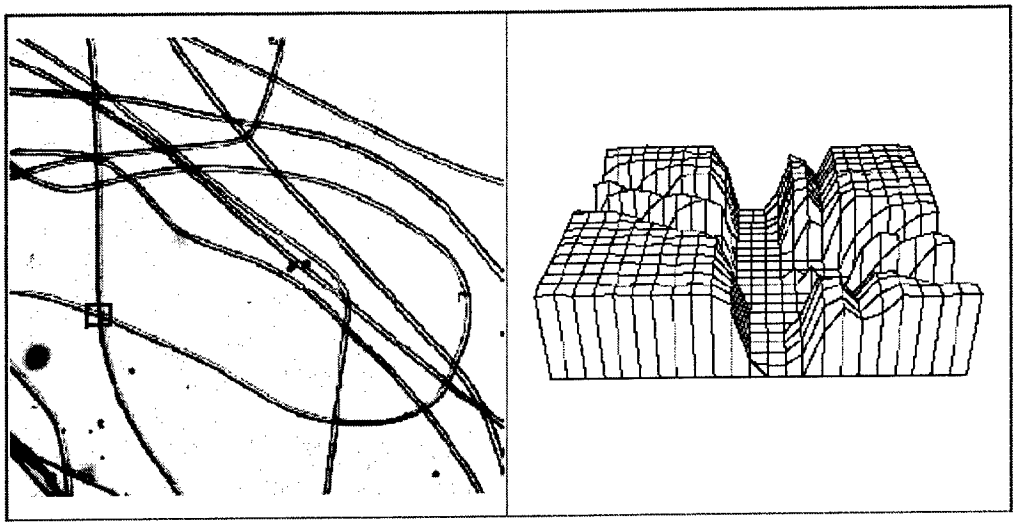


Figure 2: A fiber image and its valley topography. (a) The small rectangle marks a fiber structure in the intensity image. (b) A detail of the intensity surface marked by the rectangle in (a) reveals its valley-like topography with large minimum curvature up the valley walls and small maximum curvature along the valley floor.

of matrix types, along with methods such as determinant, cofactor, inverse, and eigen-system. The vector classes provide methods such as normalize, dot-product, and cos-angle, in addition to the standard operators such as addition, subtraction, and multiplication with a scalar. The code in the example uses an internal function `IUEi_eigensystem2` to compute the eigenvalues and vectors of a 2x2 square matrix, rather than using the matrix method, to take advantage of simplifications possible for 2x2 matrices.

Using the computed direction vector at each image location, *"non-minimal" suppression* is performed—saving only points that are local minima. The precise location of the minima may be further refined through sub-pixel interpolation. In any case, at each valley point an instance of IUE_point_edgel_2d is generated at the computed valley position and is stored for the subsequent chaining operation. The following three code fragments illustrate the non-minimal suppression function.

This first step in non-minimal suppression, shown in Example 3, interpolates the neighborhood of image values using a *multi-linear* interpolation scheme. The objective is to find intensity values along the line of maximum curvature, which is the search direction for the suppression step. The magnitude of `dirvec` determines the radius of search, which here happens to be 1.0 pixel units. The intensity value at the current location is held by `z1`; the rearward value along the curvature line is held by `z0`, while the forward value is held by `z2`.

The next step in non-minimal suppression, shown in Example 4, determines if the current pixel location is a linear minimum in terms of the image intensity function. If it is not, then the current location cannot be near a valley point. On the other hand, if it is a linear minimum, then the subsequent code performs a parabolic interpolation along the minimum curvature line using values `z0, z1, z2` respectively. This interpolation precisely locates the local minimum.

The parabolic interpolation along a line of maximum curvature is illustrated in Figure 3 for the same region that was marked by the rectangle in Figure 2(a). A detail of the intensity data with a line of maximum curvature superimposed is shown in (a). In (b), the corresponding surface cross-section with the points at z0, z1, and z2 is indicated. The computed minimum point is also labeled at the bottom of the curve.

The final step, shown in Example 5, constructs a new IUE_point_edgel_2d using the computed

```
// declarations

IUE_DOUBLE            dxx, dxy, dyy, lambda1, lambda2;
IUE_symmetric_matrix  Hessian(2);
IUE_square_matrix     Eigenvectors(2);
IUE_row_matrix        Eigenvalues(2);
IUE_vector_2d         dirvec;
IUE_INT               maxindex;
IUE_DOUBLE            NORM = 1.0 / sqrt(2.0);

// for each image location (x, y)
// compute partial derivatives . . .

    Hessian(1,1) = dxx; Hessian(1,2) = dxy;
    Hessian(2,1) = dxy; Hessian(2,2) = dyy;

    // vectors are in column space of Eigenvectors matrix
    if (!IUEi_eigensystem2(Hessian, Eigenvalues, Eigenvectors, NORM)) continue;

    lambda1 = fabs(Eigenvalues(1,1));
    lambda2 = fabs(Eigenvalues(1,2));

    if (lambda1 > lambda2)   // select largest eigenvalue
      maxindex = (lambda1 > threshold_IN) ? 1 : 0; // threshold eigenvalue
    else
      maxindex = (lambda2 > threshold_IN) ? 2 : 0;
    if (maxindex == 0) continue;

    dirvec.put_x(Eigenvectors(1,maxindex));
    dirvec.put_y(Eigenvectors(2,maxindex));

    dirvec.normalize();
```

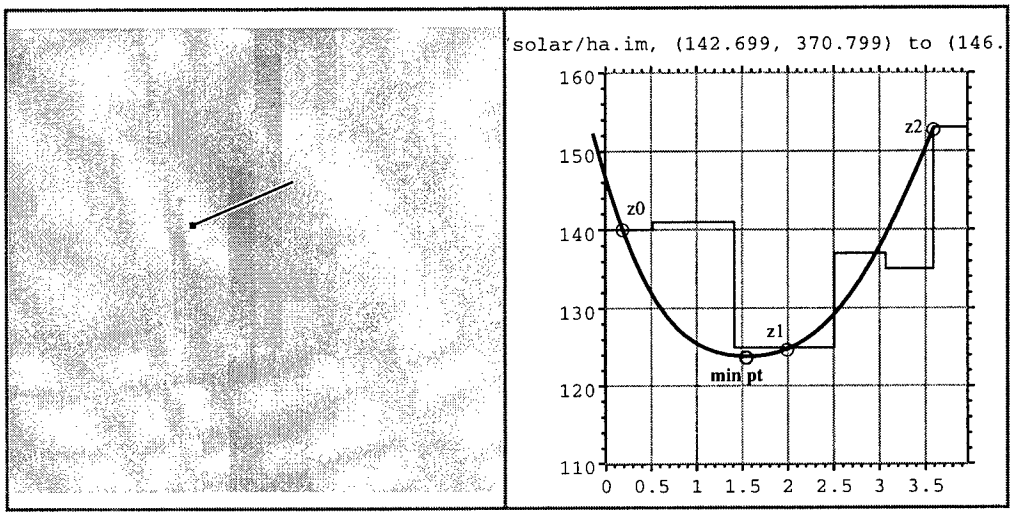
Example 2: Computing the direction of maximum curvature.

```
static
IUE_BOOL
IUEi_non_minimal_suppression
(
    Window_Type& window,
    const IUE_discrete_point_2d& pixloc,
    const IUE_vector_2d& dirvec,
    IUE_point_2d& vpt
)
{
  IUE_INT px = pixloc.x(), py = pixloc.y();
  IUE_DOUBLE    X, Y, t, z0, z2, z1 = window.get(px, py, 0, 0);
  IUE_INT       sx, sy;
  IUE_DOUBLE d1, d0;

  X = fabs(dirvec.x());
  Y = fabs(dirvec.y());
  sx = (dirvec.x() < 0) ? -1 : 1;
  sy = (dirvec.y() < 0) ? -1 : 1;

  // interpolate neighboring image values
  // assumes at least 3x3 neighborhood
  if (Y > X)
    {
      d1 = ((1-X)*window.get(px,py,0,sy))  + (X*window.get(px,py,sx,sy));
      d0 = ((1-X)*window.get(px,py,0,0))   + (X*window.get(px,py,sx,0));
      z2 = ((1-Y)*d0)                      + (Y*d1);
      d1 = ((1-X)*window.get(px,py,0,-sy)) + (X*window.get(px,py,-sx,-sy));
      d0 = ((1-X)*window.get(px,py,0,0))   + (X*window.get(px,py,-sx,0));
      z0 = ((1-Y)*d0)                      + (Y*d1);
    }
  else
    {
      d1 = ((1-Y)*window.get(px,py,sx,0))  + (Y*window.get(px,py,sx,sy));
      d0 = ((1-Y)*window.get(px,py,0,0))   + (Y*window.get(px,py,0,sy));
      z2 = ((1-X)*d0)                      + (X*d1);
      d1 = ((1-Y)*window.get(px,py,-sx,0)) + (Y*window.get(px,py,-sx,-sy));
      d0 = ((1-Y)*window.get(px,py,0,0))   + (Y*window.get(px,py,0,-sy));
      z0 = ((1-X)*d0)                      + (X*d1);
    }
```

Example 3: Non-minimal suppression: neighborhood value interpolation.

```
    // now determine if current point is a linear minimum
    // if so interpolate to get valley point: vpt
    if ((z0 <= z2 && z1 < z0) || (z2 < z0 && z1 < z2))
      {
        t = IUEi_interpolate_parabolic(z0, z1, z2);
        vpt.put_x(px + t * dirvec.x());
        vpt.put_y(py + t * dirvec.y());
        return IUE_TRUE;
      }
    else
      return IUE_FALSE;
}
```

Example 4: Non-minimal suppression: locate local minima.



(a) (b)

Figure 3: An example of parabolic interpolation. (a) The intensity data with a line of minimum curvature superimposed. (b) The corresponding surface cross-section with points z0, z1, and z2 indicated—plus, the computed minimum point.

minimum location `valley_pt` and the direction vector `dirvec`. The strength attribute is set to the largest eigenvalue and the edgel is added to the feature collection `edgel_set_OUT`.

After iterating over all image locations, the net result is a collection of IUE_point_edgel_2ds that encode the locations, directions of minimum curvature, and relative strengths of all valley points in the image. Figure 4 shows the results of running this algorithm on the fiber image. Figure 5 is a close-up, showing the edgels extracted from a portion of the image.

5 Grouping

The third fiber processing phase involves grouping the primitive data events extracted from the image into more complex structures that better represent the fibers. Grouping is an important task of intermediate-level vision. As in our example, grouping typically follows a feature extraction process and is concerned with organizing raw image features (*tokens*[1]) into coherent structures based upon apparent regularities of intrinsic, geometric, or statistical properties. Because the resulting structures integrate information over sets of tokens and make explicit significant trends in the data, they constitute a key abstraction in the overall visual computation.

Grouping can be realized by either (or both) of two complementary processing paradigms: *preattentive*, or bottom-up, grouping and *expectation-driven*, or top-down, grouping. Preattentive grouping occurs without recourse to domain-specific knowledge and aggregates tokens based upon generic regularities such as straightness, smoothness, parallelism, and symmetry, as well as similarity of color, shape, size, etc. By contrast, expectation-driven grouping exploits domain-specific world knowledge to focus the search for significant regularities among extracted features. For example, if one knows that vehicles are present in the scene and wishes to locate them, the significance of finding elliptic structure in the image is increased. Likewise with aerial images, the long ribbon-like structures of roads and runways are of increased significance. Whenever domain knowledge is available, the grouping process gains efficiency by being tuned to the specific structures.

As a computation, grouping may be characterized as computing a compatibility relation (usually higher-order) on a set of tokens. The relation expresses the particular regularity that holds for pairs, or subsets, of tokens. For example, a relation-like collinearity defined on a set of edge tokens might assemble them into groups according to their satisfaction of a collinearity (or straightness) predicate. Figure 6 illustrates this for a set of edgels, shown in the left panel. In the right panel, collinear groups of edgels are indicated by overlaying a "best-fit" line atop each group.

5.1 Representation

The grouping task requires a number of diverse representations: (a) to support the computation of the grouping relation, (b) to express the relation itself, and (c) to represent the results of the grouping process.

Computing Relations

The IUE provides two main constructs to facilitate computing grouping relations: IUE_image_feature_collection and IUE_spatial_index. An IUE_image_feature_collection is a container for a set of tokens (image-features) extracted from the same image. It supports iteration and retrieval based on attribute values. For example, it is possible to query a collection of edgel tokens for the set of all edgels with *strength* above a given threshold, λ. A collection also typically has an associated IUE_spatial_index. Spatial indices support direct retrieval of tokens based on spatial location. Since proximity is an important criterion of many grouping relations, spatial indices provide a key computational expedient.

[1]We refer to features in the data as *tokens* to avoid confusion with the C++ objects we use to represent detected features. This terminology is consistent with the literature [Marr, 1976, Stevens and Brookes, 1987].

```
is_minpt = IUEi_non_minimal_suppression(window,pixloc,dirvec,valley_pt);

if (is_minpt)
{
  tangent.put_x(-dirvec.y());
  tangent.put_y(dirvec.x());
  valley_pt += off_set;

  edgel = new IUE_point_edgel_2d(valley_pt,tangent, (IUE_scalar_image_2d*)0);

  edgel->put_strength(fabs(Eigenvalues(1, maxindex)));
  edgel_set_OUT.append(edgel);
}
```

Example 5: Constructing a point edgel.

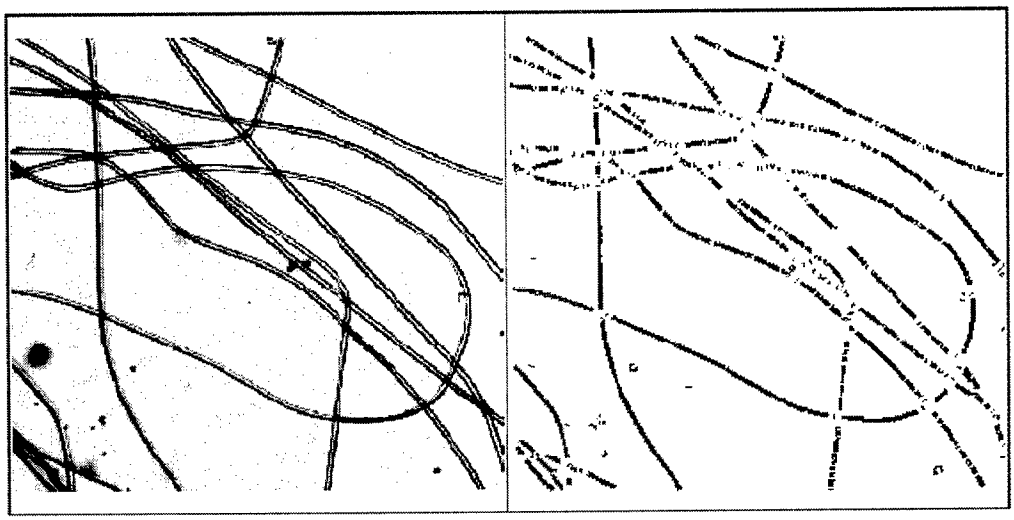


Figure 4: The valley edgels extracted from a typical fiber image. (a) The original intensity data. (b) The edgels extracted from this image.

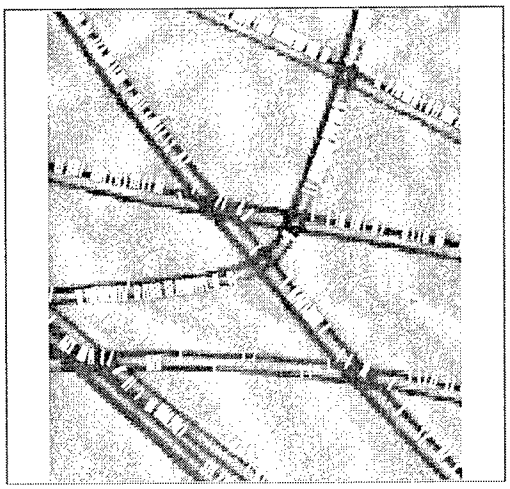


Figure 5: The valley edgels extracted from a portion of the image.

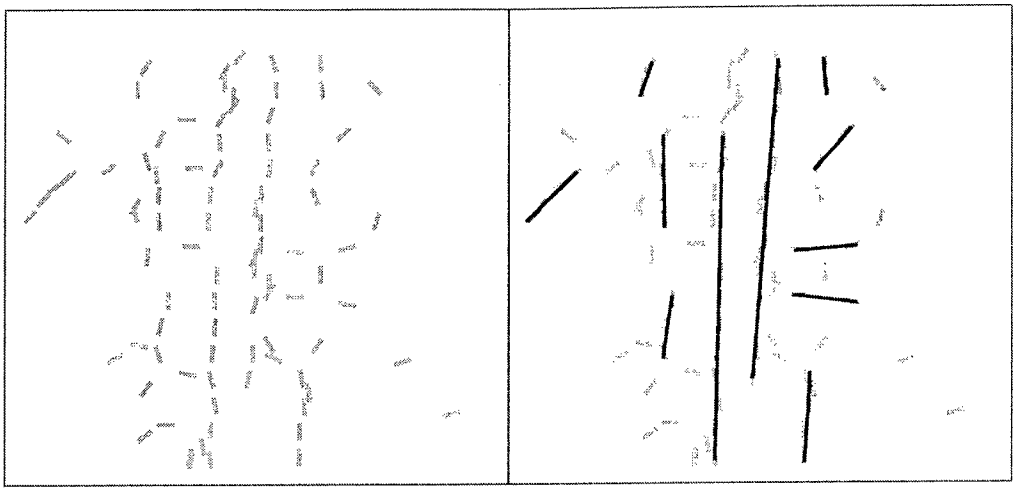


Figure 6: A set of edgels (left) and groups of edgels satisfying a collinearity predicate indicated by overlaid dark lines (right).

Expressing Relations

To represent a grouping relation, the IUE provides a set of templated graph classes: IUE_digraph_via_adj_set with its corresponding node and edge classes, IUE_digraph_vas_node and IUE_digraph_vas_edge. In the future, the class IUE_hypergraph with its corresponding edge class, IUE_hyperarc will also be available. Using the digraph classes, one can encode binary relations explicitly and higher-order relations implicitly in terms of cliques. By contrast, the hypergraph classes afford explicit representation of higher-order relations.

One can also represent grouping relations directly in terms of templated collection classes: IUE_set, IUE_sequence, and, in the future, IUE_ordered_set. The first two classes are currently available, while the last class will be added to future editions of the IUE. Using these collection classes, one can express both binary and higher-order relations. The latter two support the representation of ordered relations as well (useful for encoding relative position, importance, etc.). However, none of these collection classes directly provides the traversal and search mechanisms of graphs. This means that computations involving transitivity and closure operations may be more difficult for the user.

Representing Results

Grouping can result in either *aggregation* or *unification*. Aggregation is equivalent to assembling tokens into collections under a particular grouping relation—e.g., forming parallel sets of lines into *groups*. Unification on the other hand entails creating a new token or abstraction that embodies the essential character of the group as a whole.

An aggregation may be represented by any one of the collection classes: IUE_image_feature_collection, IUE_set, IUE_sequence, or IUE_ordered_set. One of the generic relational classes like IUE_digraph_via_adj_set or IUE_hypergraph might also provide a satisfactory representation. However the IUE also provides more specialized constructs to represent aggregations such as IUE_perceptual_group and its descendants. The IUE_perceptual_group is a form of IUE_part_instance_network that is specialized to account for a number of properties of perceptual structure like uncertainties and aggregate features.

A grouping abstraction is typically represented by a new image-feature object. For example, the collinearity of a set of edgels might be represented by an IUE_image_line_segment object (Figure 6). Effectively, the structure of the group is resolved as a single new token and, depending on the particular grouping algorithm, the new token may or may not supplant its con-

stituents.

Another useful set of classes for representing assemblages of tokens are the topology classes of the IUE. These include vertices, edges, faces, and blocks, as well as 0-, 1-, and 2-chains. These classes support the representation of complex structures in terms of their boundaries and incidence relations on their constituent tokens. Thus, a block is bounded by a set of 2-chains of faces, a face by a set of 1-chains of edges, and an edge by a 0-chain of vertices.

Quite understandably the particular classes that are used to represent the results of grouping will vary according to the structures being computed. In the examples that follow the relevant classes are IUE_point_edgel_2d, IUE_edgel_sample_2d, and IUE_edgel_chain_2d.

The class IUE_point_edgel_2d, discussed earlier in section 4.2 is one concrete variant of edgel-2d; the other is IUE_line_segment_edgel_2d. The class IUE_edgel_sample_2d is a wrapper class that supports location, tangent, and strength queries. The motivation for this class is to provide a uniform interface, from the point of view of an edgel-chain, irrespective of the particular underlying edgel class: IUE_point_edgel_2d or IUE_line_segment_edgel_2d. The data of a sample may be shared or owned by the sample. In the latter case, when the sample is deleted so is the associated data.

An IUE_edgel_chain_2d is a subclass of IUE_standard_sampled_curve_2d, which means that it is a curve described by a sequence of samples, in particular a sequence of IUE_edgel_sample_2d, and is assumed to be piecewise linear or C^0 continuous. Such a curve may be *strictly-analytic*, in which case geometric properties like tangent and curvature are ill-defined at the sample points. Or, it may not be strictly-analytic, in which case such properties are computed at sample locations from neighboring intervals of the curve—this is in order to preserve a graceful behavior throughout the curve.

5.2 Edgel Chaining

The IUE libraries currently include a number of tasks to perform grouping, including: Guy-Medioni-curves, Pavlidis-Horowitz-polylines, and Sobel-edges. For this presentation, we demonstrate a fourth IUE grouping task, an algorithm due to Glazer [1992] for grouping edgels into curvilinear structures. Essentially, the algorithm entails computing a binary relation on the collection of edgels. The connectivity defined by this relation is used to guide a chain growing process starting at the strongest ungrouped edgel. The chain is first grown in the backward direction[2] (relative to the edgel's tangent) and then in the forward direction. As edgels are added to the chain they are marked as grouped. When the chain can grow no further, a new chain is begun using the current strongest ungrouped edgel. This chaining process is accomplished via the following steps:

1. Filter the collection of edgels on the attribute *strength* to produce an *active set* A. A is an ordered set, sorted in descending order of strength. This is so that processing will proceed strongest to weakest.

2. Compute a binary relation $\mathcal{R}(A)$, based on distance and angular change as measured between pairs of edgel tokens. Formally, let $\ell_{i,j}$ be the straight line connecting tokens t_i and t_j; $d(t_i, t_j)$ the distance between t_i and t_j; β_i the angle (in absolute value) between the tangent at t_i and line $\ell_{i,j}$; and β_j the angle (in absolute value) between the tangent at t_j and line $\ell_{i,j}$. Then the pair $(t_i, t_j) \in \mathcal{R}$ iff distance $(d(t_i, t_j) < \delta_0)$ and both angles $(\beta_i, \beta_j < \beta_0)$ for user-specified thresholds δ_0 and β_0. These measures are illustrated in Figure 7(a) and (b).

3. Form a one-element edgel-chain C with the strongest remaining edgel token $t_i \in A$. If no tokens remain in A, exit; otherwise, set $t_0 = t_i$ and remove t_0 from A.

[2]This order of growing the chain, first in the backward direction and then in the forward direction, is chosen for efficiency and does not otherwise affect the algorithm.

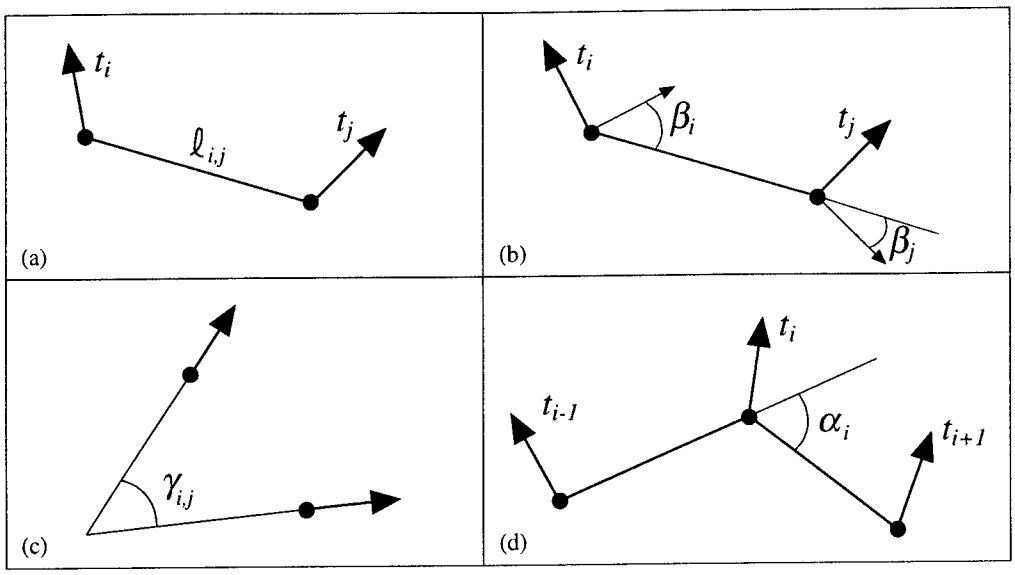


Figure 7: Link measures used for edgel chaining with Glazer algorithm. (a) The straight-line segment $\ell_{i,j}$ connecting edgel tokens t_i and t_j where $d(t_i, t_j)$ is its length. (b) The angles β_i and β_j formed between $\ell_{i,j}$ and the tangents at t_i and t_j, respectively. (c) The angle $\gamma_{i,j}$ formed between tokens t_i and t_j. (d) The turning angle α_i of the chain at token t_i.

(a) Starting with the first token t_0 of C, extend the chain backward through the strongest neighbor of t_0: $t_i \in A$, based on a measure of the *link strength* between the back end of the chain and the candidate token t_i. Set $t_0 = t_i$. Remove t_0 from A and repeat.

(b) Starting with the last token t_n of C, extend the chain forward through the strongest neighbor of t_n: $t_i \in A$, based on a measure of the *link strength* between the front end of the chain and the candidate token t_i. Set $t_n = t_i$. Remove t_n from A and repeat.

4. Go to step 3.

The measure of link strength presented in [Glazer, 1992] involves a number of terms of geometric compatibility, both binary and higher-order. The higher-order term measures the magnitude of the angle α_i between the appropriate end-tangent of the chain and the line connecting the end token to the candidate token. In the forward direction, this means the angle between the line from token t_{i-1} to t_i and the line extending from t_i through the candidate token t_{i+1}. This is shown in Figure 7(d). This angle must be small for the link-strength measure to be meaningful, i.e., $\alpha < \pi/4$. If this measure of angular compatibility between the chain and the candidate token is satisfied, then the actual link strength is computed as:

$$LS_{i,j} = d(t_i, t_j)w_d + (\beta_i + \beta_j)w_\beta + \gamma_{i,j}w_\gamma$$

where $d(t_i, t_j)$, β_i, and β_j are as defined above and shown in figures 7(a) and (b), $\gamma_{i,j}$ is the absolute value of the angle formed between tokens t_i and t_j and shown in Figure 7(c), and the ws are weights associated with each measure.

To illustrate the use of various IUE constructs, three different versions of the algorithm are presented. The first uses run-time attributes to construct an implicit graph via adjacency sets. For a set of n tokens, the graph construction requires n^2 time. Using this graph, the chaining operation requires hn expected time, where h is the average cardinality of the adjacency sets. The second version adds a spatial index to locate neighbors more efficiently. The spatial index allows the graph to be constructed in kn expected time, where k is the average density

of tokens within a disk of radius δ_0 (the maximum link radius) centered on each token. The third variant replaces the implicit graph with an explicit one. This simplifies the code and has the potential to improve the efficiency of the chaining operation because link strengths can be computed once, during the graph formation process, and then stored on each explicit graph edge.

Problem: Fiber extraction

In the previous section we discussed the detection of points corresponding to fiber locations appearing as valleys in intensity images. The result of the valley point detection process was a collection of point-edgels. Although these edgels provide evidence for the presence of fibers through local estimates of the position and direction of valley features, they are entirely disjoint. Based solely on such primitive features, one can say little about the shape and extent of the fibers themselves—indeed, it unknown which edgels are part of the same fiber. The initial task of grouping then is to determine which edgels belong to the same valley structure and to consolidate each such group into a curve. In terms of IUE classes, the input is a collection of IUE_point_edgel_2ds and the output is a collection of IUE_edgel_chain_2ds.

Figure 8 shows a collection edgels extracted from a portion of an image and the chains derived from these. Notice that the chains do not follow the entire fibers. The edgel chaining process described here groups edgels only up to junctions; the resulting set of edgel-chains therefore constitutes *fiber fragments* rather than entire fibers. A subsequent grouping step is required to recover complete fibers. This step involves many of the same concepts as edgel grouping and will not be discussed here. The individual fragments are represented as IUE_edgel_chain_2ds, and the individual edgels are overlaid using their normal vectors.

Solution (1): Fragment formation via an implicit graph

Referring to the algorithm sketched earlier in this section, the first step in the fragment formation process is to filter the initial set of edgels based upon the values of their strength attribute. This forms the set of active tokens, which is sorted in descending order of strength, allowing processing to proceed in strongest-first fashion. At this stage, a *runtime dynamic attribute* is also added to each edgel, which will allow adjacency information to be stored on the edgel, although at this point the adjacency set is empty and acts simply as a place holder.

The *dynamic attributes* mechanism of the IUE permits the user to add, at runtime, any attributes that the computation may require. Much like entries of a property list in LISP, such attributes consist of name-value pairs. And, unlike normal C++ attributes which are defined on a class for every instance of that class, dynamic attributes are defined on a per object basis. Thus, some instances of a class may have a particular runtime attribute while others do not. Further, just as with property lists, it is sufficient to do a `put` to the object with a name-value pair in order to add the attribute (property) to the object. The DAtype template provides a type safe interface to dynamic attributes. Attempting to retrieve a value as an incompatible type generates a runtime error. The code fragment in Example 6 shows how this is accomplished using the iteration macro, `IUE_FOR_EACH_ELEMENT`, to iterate over the extracted edgels. The variables `edgSETitr` and `edgSETend` are C++ Standard Template Library (STL) style iterators—incrementing the iterator moves to the next element in a collection and dereferencing an iterator returns the contents at the current position. The end iterator represents the position just beyond the last element in the collection. The `IUE_FOR_EACH_ELEMENT` macro performs the body of the loop for every element between the first and second iterators, in this case, for all elements.

Note that `DAtype< ... >::put(...)` has copy-by-value semantics; in this case each element of

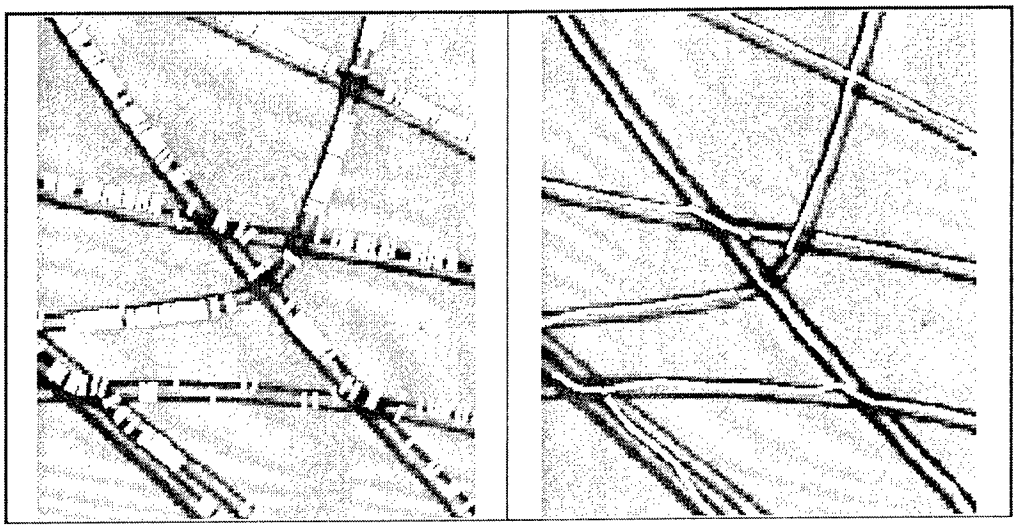


Figure 8: Chains extracted from a portion of the fiber image. (a) The extracted edgels that form the input to the grouping algorithm. (b) The chains that result.

```
  // 1) Put all sufficiently strong edgels into a sequence and sort
  // them by decreasing strength so that they can be processed
  // strongest first, initialize Active Set, and store an empty
  // adjacency set with the object

IUE_array_sequence<IUE_point_edgel_2d*>              SortedEdgels;
IUE_image_feature_collection<IUE_point_edgel_2d*>    ActiveEdgels;
IUE_image_feature_collection<IUE_point_edgel_2d*>::iterator edgSETitr;
IUE_image_feature_collection<IUE_point_edgel_2d*>::iterator edgSETend;
IUE_image_feature_collection<IUE_point_edgel_2d*>    Neighbors0;

edgSETitr = edgelset.begin();
edgSETend = edgelset.end();
IUE_FOR_EACH_ELEMENT(edgSETitr,edgSETend)
{
  edgel0 = *edgSETitr;
  if(edgel0->strength() >= MinSeedMag)
  {
    SortedEdgels.append(edgel0);
    ActiveEdgels.insert(edgel0);
    DAtype< IUE_image_feature_collection<IUE_point_edgel_2d*> >::put(
        *edgel0, "neighbors", Neighbors0);
  }
}
IUE_END_FOR_EACH_ELEMENT;

// sort is a global STL function: note parens on comparator
  sort(SortedEdgels.begin(),SortedEdgels.end(),stronger_edgel());
```

Example 6: Code to initialize the set of candidate edgels and add an empty "neighbors" collection to each edgel.

the set is copied. Thus, it is more efficient to attach an empty set `Neighbors0` to each edgel and to then directly fill this set, as opposed to first filling a temporary and then attaching a copy of the filled set to the edgel. Because both set and order semantics are required, two containers are used: the set `ActiveEdgels` and the sequence `SortedEdgels`. Each is a "reference" container in the sense that each contains pointers to edgels instead of the edgels themselves. With all sufficiently strong edgels appended to the sequence, `SortedEdgels`, the sequence is then sorted on edgel strength, as shown at the end of the example.

The function `sort` is an STL generic function that takes two iterators on an ordered container and a boolean comparator (typically user-defined) and sorts that portion the container bracketed by the two iterators. Note that the comparator `stronger` follows the STL model for comparator functions. It is an instance of the class `stronger_edgel`, a user-defined class, whose `operator ()` is a binary boolean function defined to compare the strengths of two edgels. The code fragment in Example 7 gives this definition.

The second step is to compute the binary compatibility relation on the set of active tokens. As was mentioned earlier, the relation is encoded by adding an adjacency set to each edgel. The code fragment in Example 8 illustrates how this runtime attribute is accessed and updated in the IUE.

Note that access is in terms of the attribute (property) name. A reference to the edgel's set is first obtained in `Neighbors1`, and then the function `IUEi_initial_neighbors` computes the neighbors of the current edgel under the particular binary relation and stores them in `Neighbors1`. With the completion of this code fragment, a graph of the relation is implicitly represented via the adjacency sets on the edgels.

The third step is to select the strongest token from the active set and to grow a chain both forward and backward relative to the tangent direction through the strongest ungrouped, but connected, tokens. Because an edgel chain is a sampled-curve, the `IUE_edgel_chain_2d current_chain` constructor first creates an edgel sample from edgel `*edgel0`, and from this sample, an edgel-chain consisting of a single sample is created[3]. The code fragment in Example 9 illustrates the chaining process.

Note that the seed token `edgel0` is drawn from the sorted sequence, `SortedEdgels`, so that the algorithm is *best-first*. The function `IUEi_extend_chain` grows the `current_chain` in both the forward and backward directions and removes from the `ActiveEdgels` set any edgels that are added to the chain. Chains that are sufficiently long are added to the collection of edgel-chains `ChainSet`, which is the final output of the algorithm.

Solution (2): Fragment Formation using an explicit graph

Next, we examine an alternate version of the edgel chaining algorithm, which uses an explicit rather than implicit graph to represent possible candidates. The explicit graph provides arcs between the candidates on which we can attach information concerning potential links, and supports more sophisticated linking algorithms. We will see that this allows us to avoid some duplicate computation, by saving the results on the arcs. We use this example to demonstrate graph usage. Although, in this case, the computation is simple enough that the cost of managing the graph exceeds the cost of recomputing the information.

This implementation is largely the same as the first version in terms of the measures of link strength and the order of processing. Apart from a minor addition in step 1 of the code fragments presented in Section 5.2, the primary differences between the two versions show up in step 2 (computation of the binary compatibility relation) and, more particularly, in the two helper functions: `IUEi_initial_neighbors` and `IUEi_best_neighbor`. All code that refers to the explicit graph of edgels relies on the type-

[3]The first false flag indicates that the data is shared and the second false flag indicates that the curve is not strictly analytic—cf., the spec for samples and sampled-curves.

```
class stronger_edgel :
  public binary_function<IUE_point-edgel-2d,IUE_point-edgel-2d,bool>
{
 public:
  bool operator() (const IUE_point-edgel-2d* e1,
                   const IUE_point-edgel-2d* e2) const
  {
    return(e1->strength() > e2->strength());
  }
};
```

Example 7: Comparison operator for sorting edgels according to strength.

```
// 2) For each active edgel, compute and store
// potential neighbors as a dynamic attribute of edgel
edgSETitr = ActiveEdgels.begin();
edgSETend = ActiveEdgels.end();
IUE_FOR_EACH_ELEMENT(edgSETitr,edgSETend)
{
  edgel0 = *edgSETitr;
  // get a reference to this set and fill it
  IUE_image-feature-collection<IUE_point-edgel-2d*>&
    Neighbors1 = DAtype<
                   IUE_image-feature-collection<
                      IUE_point-edgel-2d*> >::ref(
                         *edgel0, "neighbors");

  IUEi_initial_neighbors(edgel0,ActiveEdgels,Neighbors1);
}
IUE_END_FOR_EACH_ELEMENT
```

Example 8: Initialize set of neighbors for each edgel.

```
// 3) Now process edgels in descending order of strength
edgSEQitr = SortedEdgels.begin();
edgSEQend = SortedEdgels.end();
IUE_FOR_EACH_ELEMENT(edgSEQitr,edgSEQend)
{
  edgel0 = *edgSEQitr;
  if(!ActiveEdgels.is_in(edgel0)) // if already member of a chain
    continue;

  // create initial one-element chain from this edgel
  IUE_edgel-chain-2d  current_chain(*edgel0,IUE_FALSE,IUE_FALSE);
  // remove this edgel from the active set
  ActiveEdgels.remove(edgel0);

  // grow chain in both directions
  IUEi_extend_chain(current_chain,ActiveEdgels);

  // throw away short chains
  if(current_chain.size() < MinChainPts) continue;

  // add good chains to the chain collection
  ChainSet.insert((new IUE_edgel-chain-2d(current_chain)));
}
IUE_END_FOR_EACH_ELEMENT
```

Example 9: Create chains from seed edgels.

```
typedef IUE_digraph_vas_node<IUE_point_edgel_2d*,IUE_DOUBLE> EdgelNode;
typedef IUE_digraph_vas_node_abs<IUE_point_edgel_2d*,IUE_DOUBLE> EdgelNodeAbs;
typedef IUE_digraph_vas_edge<IUE_point_edgel_2d*,IUE_DOUBLE> EdgelEdge;
typedef IUE_digraph_vas_edge_abs<IUE_point_edgel_2d*,IUE_DOUBLE> EdgelEdgeAbs;
typedef IUE_digraph_via_adj_set<IUE_point_edgel_2d*,IUE_DOUBLE,
                                EdgelNode,EdgelEdge> EdgelGraph;
typedef IUE_digraph_via_adj_set_abs<IUE_point_edgel_2d*,IUE_DOUBLE> EdgelGraphAbs;
```

Example 10: Typdefs for graph classes.

defs shown in Example 10.

The key change to step 1 involves forming a node for each edgel and inserting it into the graph. This operation replaces the creation of the empty neighbors set in the previous example. The code fragment in Example 11 illustrates how this is realized.

As in the previous version, construction of the binary candidate neighbor graph is accomplished by computing the neighbors of each token by means of the function `IUEi_initial_neighbors`. However, instead of inserting candidate neighbors into sets attached to each token, the explicit graph version inserts arcs into the graph between a node and its candidate neighbors. The code fragment in Example 12 shows the relevant changes.

The code at the end of the explicit graph fragment obtains references to the nodes that correspond to the two edgels, and constructs a directed arc from `nd1` to `nd2`. Since this code is executed for every candidate edgel, every arc in one direction will have a symmetric arc in the reverse direction. This symmetry allows subsequent processing to be uniform from any node (without the necessity to special case in-arcs and out-arcs).

The function `IUEi_best_neighbor` determines which, if any, of the candidate neighbors are suitable for continuing a chain. This function computes a compatibility measure based on the information shown in Figure 7. In the first version, some of this information (`L2, beta1, beta2`) was already computed in `IUEi_initial_neighbors` to initialize the adjacency sets. But, because these terms were not stored, they must be recomputed each time. With an explicit graph, however, we can save this information on the arcs between the candidates, and avoid recomputing it. The code fragment from `IUEi_initial_neighbors` shown in Example 12 records this information as `link_strength` in the call to `create_edge`.

The code fragment in Example 13 shows the body of the original version of `IUEi_best_neighbor`.

The differences between the two versions of this function are most apparent by comparing their bodies. Using an explicit graph, the first step in computing the best neighbor of an edgel is to retrieve the set of graph edges incident on the node associated with edgel `e1` as displayed in Example 14.

Now in iterating over the out edges of this node, it is first necessary to check whether the edgel associated with each neighboring node is still active. We use the label of the node to get to the edgel and see if the edgel is still in the `Active` set, as shown in Example 15.

With an explicit graph, the link strengths are stored as labels on the graph edges so it is not necessary to recompute them each time. Example 16 fragment shows that the turn angle is measured in the same way as in the previous version but that link strength is simply retrievable as a stored attribute.

The code fragment in Example 17 shows the complete body of the new version of `IUEi_best_neighbor`.

Solution (3): Fragment Formation using a spatial-index

Finally, we examine an improved version of the edgel chaining algorithm, which uses a spatial index to locate candidate edgels rather than computing a graph. Much of the implementation is the same as the previous versions. However, we move all compatibility computation into the `IUE_best_neighbor` function and remove the use of `IUE_initial_neighbors`.

The first step in this version is to "paint" the candidate edgels into a spatial index. The code fragment in Example 18 constructs the spatial index and associates it with the collection of active edgels. It is unnecessary to explicitly "paint" the edgels into the spatial index since the image feature collection class takes care of painting the features into its associated spatial index as they are added to the collection. Thus, once we associate the index with the collection, the code to add edgels to the collection is unchanged.

The min and max values are computed either

```
IUE_FOR_EACH_ELEMENT(edgSETitr,edgSETend){
  edgel0 = *edgSETitr;
  if(edgel0->strength() >= MinSeedMag){
    SortedEdgels.append(edgel0);
    ActiveEdgels.insert(edgel0);

    // create a graph node for this edgel
    theEdgelGraph.create_node(edgel0);
  }
}
IUE_END_FOR_EACH_ELEMENT;
```

Example 11: Constructing a graph node for each token.

```
                    With implicit graph (version 1)

IUE_FOR_EACH_ELEMENT(edgiter,edgend){
    e2 = *edgiter;
    // determine if link length and angle is appropriate.  If not, try next token...

    // if compatibility satisfied, insert e2 into the e1's collection of neighbors
    Neighbors.insert(e2);
  }
IUE_END_FOR_EACH_ELEMENT

                    With explicit graph (version 2)

IUE_FOR_EACH_ELEMENT(edgiter,edgend){
    e2 = *edgiter;
    // determine if link length and angle is appropriate.  If not, try next token...

    // if compatibility satisfied, insert arc between e1 and e2
    // and label it with link_strength
    // get nodes associated with the edgels
    nd1 = theEdgelGraph.assoc_node(e1);
    nd2 = theEdgelGraph.assoc_node(e2);
    // insert edge between e1 and e2
    theEdgelGraph.create_edge(link_strength, *nd1, *nd2);
}
IUE_END_FOR_EACH_ELEMENT
```

Example 12: Constructing the candidate neighbors graph: implicit (top) and explicit (bottom).

```
// get neighbors of seed edgel: e1
IUE_image_feature_collection<IUE_point_edgel_2d*>&
  Candidates = DAtype< IUE_image_feature_collection<IUE_point_edgel_2d*> >::
                          ref(*e1,  "neighbors");
edgiter = Candidates.begin();
edgend = Candidates.end();

IUE_FOR_EACH_ELEMENT(edgiter,edgend){
  e2 = *edgiter;
  if(!Active.is_in(*e2)) continue;      // consider only Active edgels

  // measure link strength with each candidate...

  alpha = IUEi_compute_turn_angle(e1,e2,chain0,forward);
  if(alpha >= MaxTurnAngle) continue;

  // candidates satisfy conditions on next 2 measures
  // but must recompute anyway because link_strength is not stored
  L2 = IUEi_compute_link_length(e1,e2);
  IUEi_compute_edgel_to_link_angles(e1,e2,beta1,beta2);
  gamma = IUEi_compute_edgel_to_edgel_angle(e1,e2);

  link_strength = L2 * LinkLengthWeight +
    (beta1 + beta2) * EdgelToLinkAngleWeight +
    gamma * EdgelToEdgelAngleWeight;

  // save best one
  if(link_strength > max_strength){
    best_edgel = e2;
    max_strength = link_strength;
    found = IUE_TRUE;
  }
}
IUE_END_FOR_EACH_ELEMENT
return(found);
```

Example 13: Original chaining function (implicit graph).

```
// get edges from the seed edgel: e1
EdgelNode *nd1 = theEdgelGraph.assoc_node(e1);

// now construct iterators on the set of out edges
edgiter = nd1->edges().begin();
edgend = nd1->edges().end();

// finally iterate over edges to find the best...
```

Example 14: Retrieve arcs incident on a node in the graph.

```
// iterate over edges to find the strongest
IUE_FOR_EACH_ELEMENT(CandEdgeIter,CandEdgeEnd){
  edge0 = *CandEdgeIter;

  // get the neighboring edgel
  e2 = edge0->to_node()->label();
  if(!Active.is_in(e2)) continue;     // consider only Active edgels

  // measure turn angle and retrieve link strength...

  // as before, record strongest edgel
}
  IUE_END_FOR_EACH_ELEMENT
```

Example 15: Obtaining candidate edgel from graph.

```
// turn angle with this edgel must not be too great
alpha = compute_turn_angle(e1,e2,chain0,forward);
if(alpha >= MaxTurnAngle) continue;

// test link strength to find the strongest neighboring edgel
link_strength = edge0->label();
```

Example 16: Obtaining `link_strength` value from label of arc.

```
// get neighbors of seed edgel: e1
EdgelNode *nd1 = theEdgelGraph.assoc_node(e1);
edgiter = nd1->edges().begin();
edgend = nd1->edges().end();

IUE_FOR_EACH_ELEMENT(edgiter,edgend){
  // get the neighboring edgel
  e2 = edge0->to_node()->label();
  if(!Active.is_in(e2)) continue;     // consider only Active edgels

  // turn angle with this edgel must not be too great
  alpha = IUEi_compute_turn_angle(e1,e2,chain0,forward);
  if(alpha >= MaxTurnAngle) continue;

  // retrieve link strength
  link_strength = edge0->label();

  // save best one
  if(link_strength > max_strength){
    best_edgel = e2;
    max_strength = link_strength;
    found = IUE_TRUE;
  }
}
IUE_END_FOR_EACH_ELEMENT
return(found);
```

Example 17: Chaining function using an explicit graph.

```
// Create the spatial index
IUE_INT x_low  = (IUE_INT)min_x - 1;
IUE_INT x_high = (IUE_INT)max_x + 2;
IUE_INT y_low  = (IUE_INT)min_y - 1;
IUE_INT y_high = (IUE_INT)max_y + 2;
IUE_array_spatial_index_2d *ArraySI =
    new IUE_array_spatial_index_2d(x_low, y_low, x_high, y_high, 10, 10);

// Associate the spatial index with the image feature collection
ActiveEdgels.put_index(ArraySI);
```

Example 18: Adding a spatial index to an image feature collection.

by inspecting the edgels or from knowledge of the image coordinates. The final two arguments to the constructor call specify the resolution of the index.

The payoff for using the spatial index occurs when computing the set of candidate token pairs. In the previous versions, the function `IUEi_initial_neighbors` performed this computation by considering every possible neighbor, using an $O(n^2)$ algorithm. For each pair of tokens the function would measure the distance between tokens `e1` and `e2` and the compute the link angles formed by these tokens (Figure 7(a) and (b)). If these measures are less than the given thresholds `MaxLinkLength` and `MaxEdgelToLinkAngle`, token `e2` is added to the adjacency set of token `e1`.

In the new version, we instead use a spatial index to locate candidate neighbors instead of an adjacency set. Such an index allows tokens to be prehashed and retrieved based on their spatial locations. The spatial index allows the user to retrieve only those tokens within a certain distance of a given point (in this case, within `MaxLinkLength` of token `e1`) by means of the method `radial_fetch`. This is useful as one of the criteria of the compatibility relation is distance and the resulting set of `Candidates` is, in general, much smaller than the active set used in the first version.

Since the spatial index provides a efficient means to locate these candidates, we can use the index directly in the `IUEi_best_neighbor` function and avoid `IUEi_initial_neighbors` altogether.

The code fragment in Example 19 shows the new version of `IUEi_best_neighbor`.

Comparing this version with `IUEi_best_neighbors` shown in Example 13 we see that the new version iterates over the tokens in `Active` that are within a radius of `MaxLinkLength` of the current token, rather than all of the tokens in the adjacency set. This is immediately beneficial, since the spatial index contains only active elements. Thus, the new version computes the compatibility measure only for tokens that have not already been linked, whereas in the previous version, this measure was computed for all nearby candidates.

5.3 Performance

Figure 9 shows the relative performance of the three grouping algorithms we discussed in this section. The righthand columns show the time spent in the various stages of the algorithm. Note that the first two algorithms spend most of their time computing the adjacency sets. The spatial index algorithm spends more time computing the best neighbor for each edgel in a chain than the other two, since all of the compatibility computation occurs there. However, this extra work per comparison is offset by requiring many fewer comparisons, since it only compares tokens that are not already linked into chains. The difference in the number of comparisons is shown in Figure 10. The far left column in the graph depicts the total execution time for the grouping tasks.

6 Application program

The end result of programming with the IUE is either a task or an application. A *task* is basically an algorithm implementation that is packaged into a single function. To be an IUE task, it must conform to IUE specifications so that the IUE's task-related tools can properly handle the task. These specifications currently consist of three components:

1. A function interface that meets the following requirements:
   - All output must occur through pass-by-reference arguments, rather than by return type.
   - All arguments must be pass-by-value or pass-by-reference.
   - Templated types must be replaced with typedefs.
2. An interface description file (an *fdf* or *function description file*) that describes the in-

```
// get candidates nearby seed edgel: e1
IUE_set_abs<IUE_spatial_object*>
  *Candidates = Active.index()->radial_fetch(*e1, MaxLinkLength);

// construct iterators on the candidate set
IUE_set_abs<IUE_spatial_object*>::iterator edgiter, edgend;
edgiter = Candidates->begin();
edgend = Candidates->end();

IUE_FOR_EACH_ELEMENT(edgiter,edgend){
  e2 = (IUE_point_edgel_2d*)*edgiter;

  // turn angle with this edgel must not be too great
  alpha = IUEi_compute_turn_angle(e1,e2,chain0,forward);
  if(alpha >= MaxTurnAngle) continue;

  // determine if link length and angle is appropriate
  L2 = IUEi_compute_link_length(e1,e2);
  if(L2 >= MaxLinkLength) continue;

  IUEi_compute_edgel_to_link_angles(e1,e2,beta1,beta2);
  if(beta1 >= MaxEdgelToLinkAngle
     || beta2 >= MaxEdgelToLinkAngle) continue;

  gamma = IUEi_compute_edgel_to_edgel_angle(e1,e2);

  link_strength = L2 * LinkLengthWeight +
    (beta1 + beta2) * EdgelToLinkAngleWeight +
    gamma * EdgelToEdgelAngleWeight;

  // save best one
  if(link_strength > max_strength){
    best_edgel = e2;
    max_strength = link_strength;
    found = IUE_TRUE;
  }
}
IUE_END_FOR_EACH_ELEMENT
delete Candidates;
return(found);
```

Example 19: IUEi_best_neighbor using a spatial index.

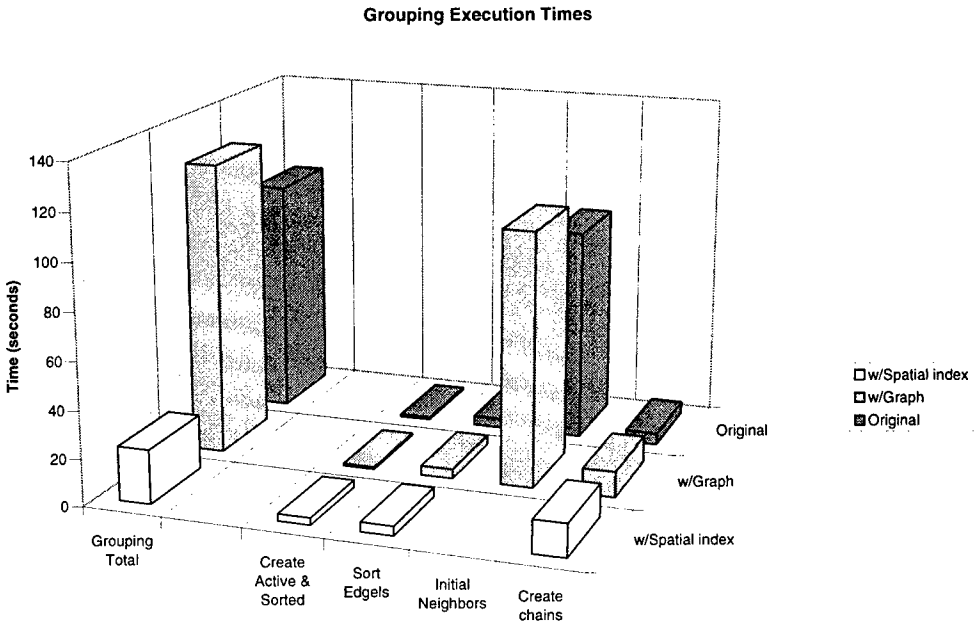


Figure 9: Execution times for the three grouping tasks.

	Original	With graph	With spatial index
Calls to Initial Neighbor	3515	3515	0
Calls to Best Neighbor	3732	3732	3732
Initial Neighbor Comparisons	12M	12M	0
Best Neighbor Comparisons	100K	100K	58K

Figure 10: Number of comparisons performed by the three grouping tasks.

terface to the IUE tools[4].

3. Documentation in the form of an IUE task specification document.

The IUE currently provides tools to generate task documentation from the specification files, and to generate a Khoros glyph wrapper. The documentation tools generate documentation that has the same look-and-feel as the existing task documentation, including automatic generation of HTML versions of the documentation. The Khoros wrapping tool, `gen_kroutine`, provides the means to place the task in a Khoros toolbox which enables users to drop the task onto a Cantata desktop and link it to other Khoros and IUE task glyphs.

An application is simply a C++ program that invokes the task functions defined in some library, or a Cantata desktop that links tasks to perform some function.

6.1 Fiber extraction tasks

The previous sections described three tasks: an image processing task from the IUE library (`gaussian-filter`), a feature extraction task, and a grouping task. The code fragment in Example 20 shows how these tasks can be linked together into a simple application.

The `IUE_image_pointer` class allows images to be passed as a reference to pointer, and allows the Khoros tools to provide special handling of image arguments.

6.2 Visualization

Image Understanding algorithms are rather difficult to develop without effective tools to graphically visualize results. The IUE addresses this problem by providing both a visualization library and stand-alone tools. The library allows algorithm developers to integrate visualization into their applications. The stand-alone tools allow users to display and manipulate data written out to IUE Data Exchange (DEX) files.

[4]We expect to generate the fdf automatically or implicitly in a future IUE release.

The IUE currently provides two visualization tools: an X-based tool, built upon the Fresco and MesaGraphics user interface libraries, and a Java-based tool that can access data over a network and act as a client to an IUE server program.

For the former, the C++ visualization library, called *Parmesan*, implements a display as a set of overlaid planes containing spatially registered data. A *DataManager* window allows users to select the data sets to display, alter their stacking order, and access their properties. Parmesan currently provides a very simple interface that allows developers to write code to bring up a display and add data sets. The code fragment in Example 21 shows the Parmesan calls necessary to display the intermediate and final results of the fiber application. Future improvements should allow Parmesan to access pre-loaded image data, rather than requiring files, and provide a separate thread of control so users can interact with the display while the application is running, rather than waiting until the end.

Figure 11 shows the final fiber application display. The righthand view shows both the edgels and chains. Once the application has complete, the user can select which overlays to view, and create new views, as shown in Figure 12.

7 Conclusions

The IUE provides an expressive representation hierarchy that covers most low and intermediate level image understanding concepts. We have demonstrated and discussed, using a fiber extraction program, a number of areas of the IUE and its class hierarchy, including:

- base classes: sets, matrices, graphs, and spatial indices
- image-features and spatial-objects: points, edgels, image-feature-collections, edgel-chains, and curves
- IUE tasks: image processing (gaussian-filter), feature extraction (Extremal-curvature, valley-ridges), and grouping (GlazerChains)

```
// Obtain inputs and parameters...

// Read an image from a file and create an image pointer
IUE_scalar_image_2d* im_in = read_image( input_image );
IUE_image_pointer im_ptr_in, im_ptr_out;
im_ptr_in.put_image_ptr(im_in);

// Apply a filter to smooth the input image
AAI_gaussian_smooth(im_ptr_in, order_x, order_y, im_ptr_out);

// Extract edgels from the smoothed image
IUE_image_feature_collection<IUE_edgel_2d*> edgels;
IUEt_valley_ridges(im_ptr_out, area_of_interest,
                   minimum_curvature, curvature_ratio,
                   feature_type, 1, edgels);

// Link the edgels into chains
IUE_image_feature_collection<IUE_edgel_chain_2d*> chainSet;
IUEt_chain_edgels(edgels, MinSeedMag, MinChainPts, MaxLinkLen,
                  MaxEdgel2LinkAng, MaxTurnAng, LinkLenWt,
                  Edgel2LinkAngWt, Edgel2EdgelAngWt, chainSet);
```

Example 20: Fiber extraction application code

- visualization: Parmesan library, standalone C++ display, Java display

Additional papers[Dolan *et al.*, 1996, Kohl *et al.*, 1995, Kohl *et al.*, 1994] discuss other important parts of the IUE, including:

- additional base classes: arrays and sequences
- coordinate systems and transforms: cartesian, geographic, color
- image classes: scalar, RGB, tuple, support for very large images, image accessors and filters
- sensors and sensor models
- statistics: histograms

7.1 Obtaining the IUE

The IUE Core consists of the IUE class library, including complete specification and sources, HTML and PostScript documentation, a primer, and support libraries. In addition, pre-compiled libraries are available for our supported architectures: SunOS4, Solaris, and Linux2. The IUE is available via anonymous FTP from Amerinex and a number of mirror sites in the US, Canada, Europe, and Japan. To get full information on ftp and web access to the IUE, send email to iue-info@aai.com with the subject "HELP", or visit Amerinex's web site at http://www.aai.com. To join the iue-users mailing list, send email to iue-users-request@aai.com.

Acknowledgments

We wish to thank Mark Roubentchik and Scott Irwin for their work implementing and testing the examples in this paper, and the rest of IUE team, past and present, for their all of their contributions to the IUE.

References

[Barrow and Tenenbaum, 1978] H. Barrow and J. Tenenbaum. Recovering intrinsic scene characteristics from images. In A. Hanson

```
    // Obtain inputs and parameters...

// Initialize Parmesan
IUE_parmesan::init(argc,argv);

    // Read an image from a file and create an image pointer
    IUE_scalar_image_2d* im_in = read_image( input_image );
    IUE_image_pointer im_ptr_in, im_ptr_out;
    im_ptr_in.put_image_ptr(im_in);

// Create a new display and show the input image
IUE_parmesan::newWindow("Grouping");
IUE_parmesan::appendImage(input_image, "Input image");
IUE_parmesan::run_one_event();

// Scale the display to fit the image in the window
IUE_parmesan::normalize_display(im_in->x_size(), im_in->y_size());
IUE_parmesan::run_one_event();

    // Apply a filter to smooth the input image
    AAI_gaussian_smooth(im_ptr_in, order_x, order_y, im_ptr_out);

// Display the filtered image
write_image("Smoothed image", im_ptr_out, gauss_image);
IUE_parmesan::appendImage(gauss_image, "Smoothed image");
IUE_parmesan::run_one_event();

    // Extract edgels from the smoothed image
    IUE_image_feature_collection<IUE_edgel_2d*> edgels;
    IUEt_valley_ridges(im_ptr_out, area_of_interest,
                       minimum_curvature, curvature_ratio,
                       feature_type, 1, edgels);

// Display the edgels
IUE_parmesan::append(&edgels, "Valley Edgels");
IUE_parmesan::run_one_event();

    // Link the edgels into chains
    IUE_image_feature_collection<IUE_edgel_chain_2d*> chainSet;
    IUEt_chain_edgels(edgels, MinSeedMag, MinChainPts, MaxLinkLen,
                      MaxEdgel2LinkAng, MaxTurnAng, LinkLenWt,
                      Edgel2LinkAngWt, Edgel2EdgelAngWt, chainSet);

// Display the chains
IUE_parmesan::append(&ChainSet, "Glazer Chains");
IUE_parmesan::run_one_event();

// Allow the user to interact with the display
cout << "\n\n Type Ctrl-Z to continue." << endl;
IUE_parmesan::run();
```

Example 21: Parmesan visualization code added to fiber extraction application

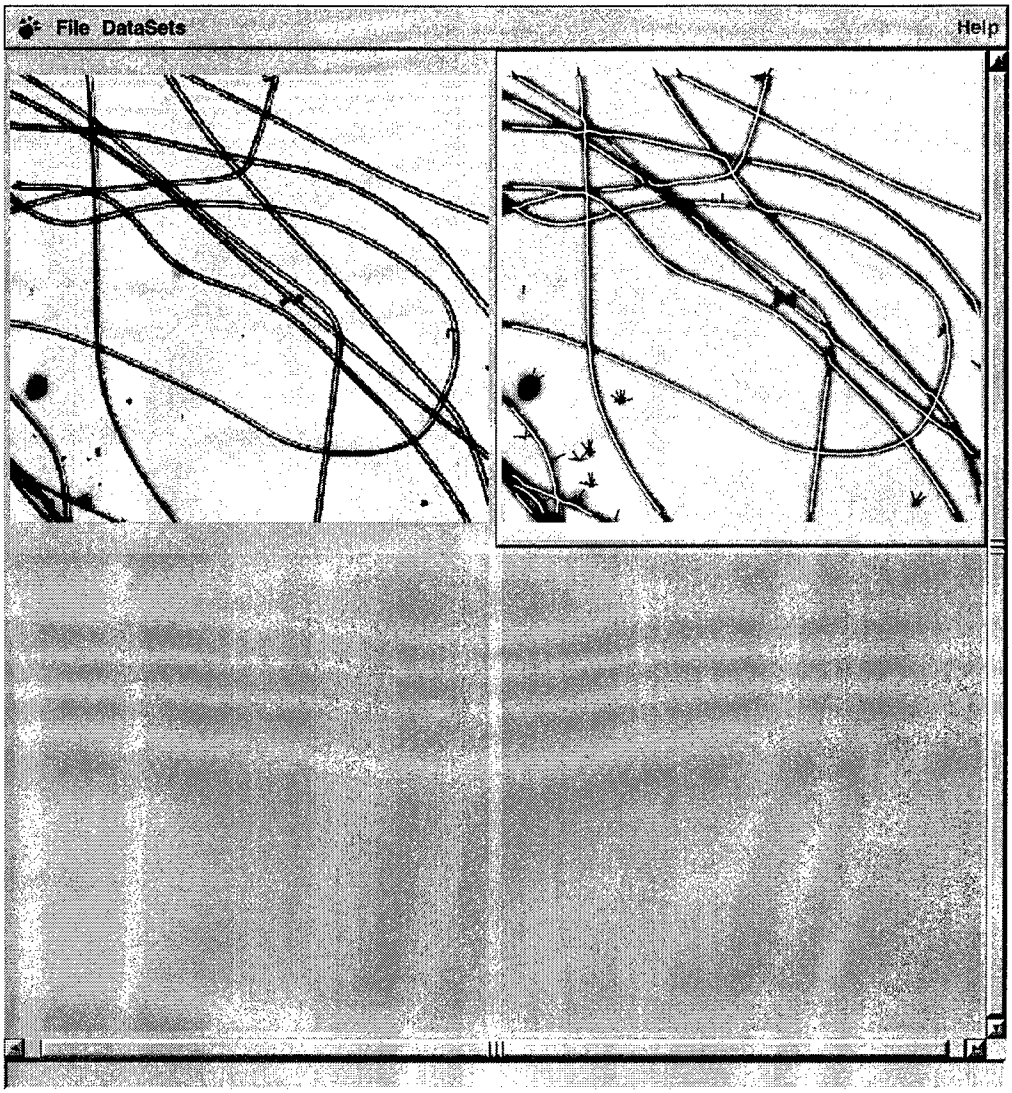


Figure 11: Display generated by fiber application. The top left pane shows the input image. The top right pane shows the smoothed image with overlays of edgels and chains.

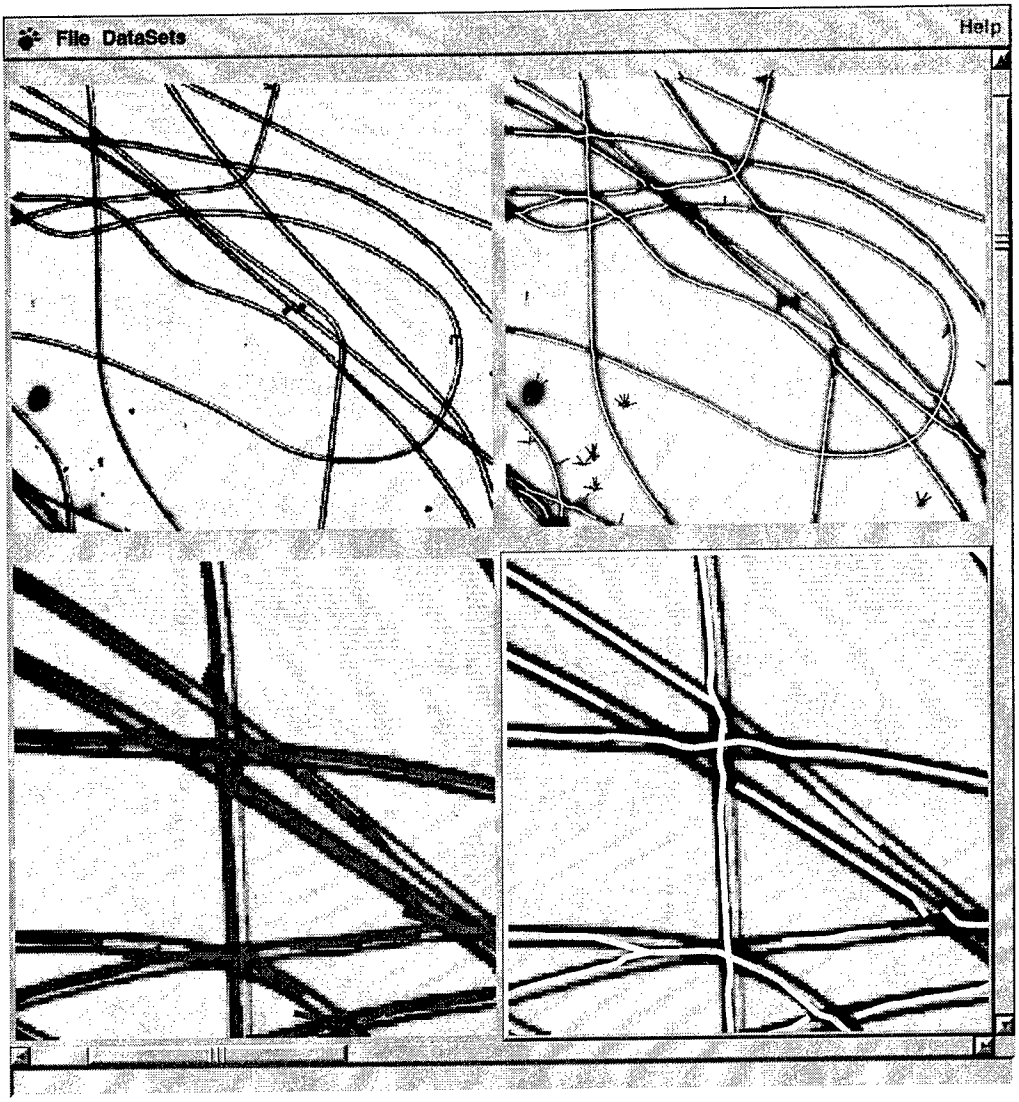


Figure 12: Fiber display after user interactively created additional views of the data.

and E. Riseman, editors, *Computer Vision Systems*, pages 3–26. Academic Press, New York, 1978.

[Dolan *et al.*, 1996] J. Dolan, C. Kohl, R. Lerner, J. Mundy, T. Boult, and J. R. Beveridge. Solving diverse image understanding problems using the image understanding environment. In *Proc. ARPA IUW*, 1996.

[Glazer, 1992] F. Glazer. Fiber identification in microscopy by ridge detection and grouping. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pages 205–212, Palm Springs, CA, 1992.

[Kohl *et al.*, 1994] C. Kohl, R. Lerner, A Hough, C. Loiselle, J. Dolan, M. Friedman, and M. Roubentchik. A stellar application of the IUE: Solar feature extraction. In *Proc. DARPA IUW*, 1994.

[Kohl *et al.*, 1995] C. Kohl, J. J. Hunter, and C. Loiselle. Towards a unified IU environment: Coordination of existing IU tools with the iue. In *Proceedings of the 5th International Conference on Computer Vision*, 1995.

[Marr, 1976] D. Marr. Early processing of visual information. *Phil. Trans. of Royal Society of London*, 275:483–519, 1976.

[Stevens and Brookes, 1987] K. Stevens and A. Brookes. Detecting structure by symbolic constructions on tokens. *CVGIP*, 37:238–260, 1987.