ADA162 898
GC970145

UNCLASSIFIED

Technical Report
distributed by

DEFENSE TECHNICAL INFORMATION CENTER

Defense Logistics Agency
Defense Technical Information Center
Cameron Station
Alexandria, Virginia 22304-6145

UNCLASSIFIED

UNCLASSIFIED

NOTICE

We are pleased to supply this document in response to your request.

The acquisition of technical reports, notes, memorandums, etc., is an active, ongoing program at the **Defense Technical Information Center (DTIC)** that depends, in part, on the efforts and interest of users and contributors.

Therefore, if you know of the existence of any significant reports, etc., that are not in the **DTIC** collection, we would appreciate receiving copies or information related to their sources and availability.

The appropriate regulations are Department of Defense Directive 3200.12, DoD Scientific and Technical Information Program; Department of Defense Directive 5230.24, Distribution Statements on Technical Documents *(amended by Secretary of Defense Memorandum, 18 Mar 1984, subject: Control of Unclassified Technology with Military Application)*; American National Standard Institute *(ANSI)* Standard Z39.18, Scientific and Technical Reports: Organization, Preparation, and Production; Department of Defense 5200.1R, Information Security Program Regulation.

Our Acquisition Section, **DTIC-FDAB**, will assist in resolving any questions you may have. Telephone numbers of that office are:
(202) 274-6847, (202) 274-6874 or Autovon **284-6847, 284-6874.**

DO NOT RETURN THIS DOCUMENT TO DTIC

EACH ACTIVITY IS RESPONSIBLE FOR DESTRUCTION OF THIS DOCUMENT ACCORDING TO APPLICABLE REGULATIONS.

UNCLASSIFIED

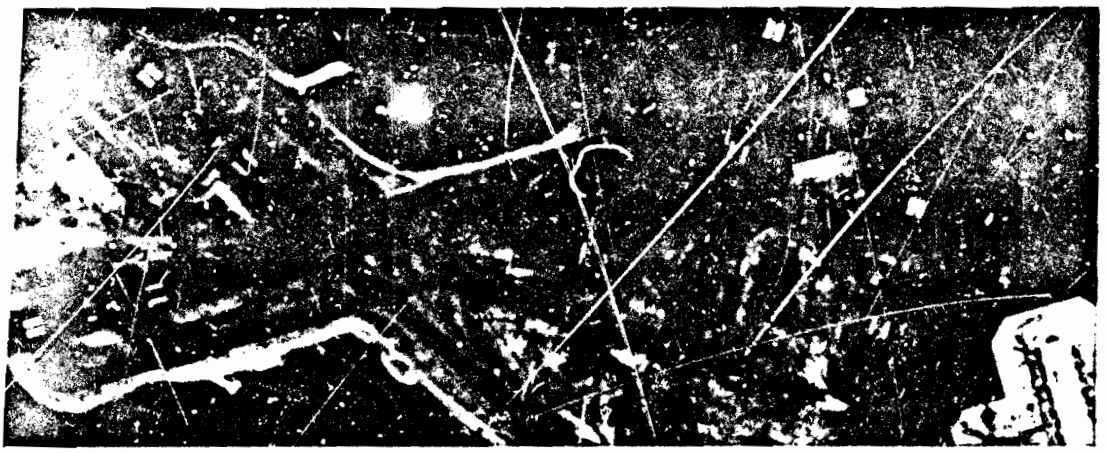
(12)

IMAGE UNDERSTANDING WORKSHOP

DECEMBER 1985

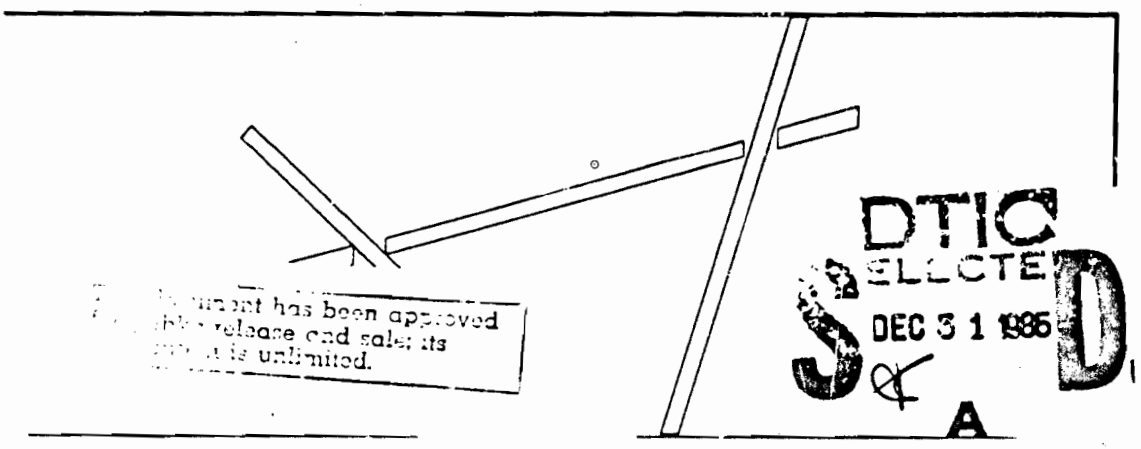
Sponsored by:
Information Processing Techniques Office
Defense Advanced Research Projects Agency

AD-A162 898



INPUT
(BOSTON LOGAN AIRPORT)

DTIC FILE COPY



DTIC
ELECTE
DEC 3 1 1985
S D
A

FINAL OUTPUT
(DETECTED RUNWAYS)

...ument has been approved
...release and sale; its
...n is unlimited.

85 12 30 102

SAIC™
Science Applications International Corporation

(12)

IMAGE UNDERSTANDING

Proceedings of a Workshop
Held at
Miami Beach, Florida
December 9-10, 1985

Sponsored by the
Defense Advanced Research Projects Agency

Science Applications International Corporation
Report Number SAIC-85/1149
Lee S. Baumann
Workshop Organizer and
Proceedings Editor

This report was supported by
The Defense Advanced Research
Projects Agency under DARPA
Order No. 3456, Contract No. MDA903-84-C-0160
Monitored by the
Defense Supply Service, Washington, D.C.

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION LIMITED

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 United States Government.

DTIC
ELECTE
DEC 3 1 1985
S
A

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: SAIC-85/1149	2. GOVT ACCESSION NO.: AD-A162 898
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): IMAGE UNDERSTANDING Proceedings of a Workshop, December 1985	5. TYPE OF REPORT & PERIOD COVERED: ANNUAL TECHNICAL October 1984 - December 1985
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): LEE S. BAUMANN (Ed.)	8. CONTRACT OR GRANT NUMBER(s): MDA903-84-C-0160
9. PERFORMING ORGANIZATION NAME AND ADDRESS: SCIENCE APPLICATIONS INTERNATIONAL CORPORATION 1710 Goodridge Drive, 10th Floor McLean, Virginia 22102	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: ARPA ORDER No. 3456
11. CONTROLLING OFFICE NAME AND ADDRESS: Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209	12. REPORT DATE: December 1985
	13. NUMBER OF PAGES: 515
14. MONITORING AGENCY NAME & ADDRESS (If different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital Image Processing; Image Understanding; Scene Analysis; Edge Detection; Image Segmentation; CCDArrays; CCD Processors.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document contains the outlines of annual progress reports and technical papers presented by the research activities in Image Understanding, sponsored by the Information Processing Techniques Office; Defense Advanced Research Projects Agency. The papers were presented at a workshop conducted on 9-10 December 1985, in Miami Beach, Florida. Also included are copies of invited papers presented at the workshop and additional technical papers from the research activities which were not presented due to lack of time but are germane to this research field.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

	Page
FOREWORD	i
DEFENSE TECHNICAL INFORMATION CENTER ACCESSION NUMBERS	v
AUTHOR INDEX	vi

SECTION I - PROGRAM REVIEWS BY PRINCIPAL INVESTIGATORS

"Image Understanding Techniques for Autonomous Vehicle Navigation", A. Rosenfeld and L.S. Davis; University of Maryland	1
"Image Understanding Research at CMU", T. Kanade and S. Shafer; Carnegie-Mellon University	4
"Image Understanding Research at USC: 1984-85", R. Nevatia; University of Southern California	10
"Recent Progress of the Rochester Image Understanding Project", J.A. Feldman and C.M. Brown; University of Rochester	16
"Spatial Understanding", T.O. Binford; Stanford University	20
"MIT Progress in Understanding Images", T. Poggio and the staff; Massachusetts Institute of Technology	25
"The SRI Image Understanding Research Program", M.A. Fischler; SRI International	40
"Image Understanding Research at Columbia", J.R. Kender; Columbia University	45
"Summary of Progress in Image Understanding at the University of Massachusetts", E.M. Riseman and A.R. Hanson; University of Massachusetts	48

QUALITY INSPECTED 3

Availability Codes

Avail and/or Special

A-1

TABLE OF CONTENTS

Page

SECTION II - INVITED TECHNICAL REPORTS

"Knowledge-Based Interpretation Aids to the Navy Oceanographic Image Analyst", LCDR J.D. McKendrick and M. Lybanon; Naval Ocean Research and Development Activity 61

"The ESPI Vision System", T.C. Rearick; Lockheed-Georgia Company 64

"Dynamic Archival Scene Model", H.N. Nasr, R.K. Aggarwal and D.P. Panda; Honeywell Inc. 74

"Image Understanding Research at General Electric", J.L. Mundy; General Electric 83

"Structure and Motion From Images", J.K. Aggarwal and A. Mitiche; The University of Texas at Austin 89

"Surfaces From Stereo", W. Hoff and N. Ahuja; University of Illinois 98

SECTION III - TECHNICAL REPORTS PRESENTED

"Structure from Motion Without Correspondence: General Principle", K. Kanatani; University of Maryland 107

"Robust Estimation of 3-D Motion Parameters from a Sequence of Image Frames Using Regularization", G. Medioni and Y. Yasumoto; University of Southern California 117

"Contour, Orientation and Motion", J. Aloimonos, A. Basu and C.M. Brown; University of Rochester 129

"Epipolar-Plane Image Analysis: A Technique for Analyzing Motion Sequences", R.C. Bolles and H.H. Baker; SRI International 137

TABLE OF CONTENTS

Page

SECTION III - TECHNICAL REPORTS PRESENTED (Continued)

"SRI's Baseline Stereo System", M.J. Hannah; SRI International 149

"Concurrent Multilevel Relaxation", D. Terzopoulos; Massachusetts Institute of Technology 156

"Trinocular Vision Using Photometric and Edge Orientation Constraints", V.J. Milenkovic and T. Kanade; Carnegie-Mellon University 163

"Edgel-Aggregation and Edge-Description", V.S. Nalwa and E. Pauchon; Stanford University 176

"Introducing a Smoothness Constraint in a Matching Approach for the Computation of Displacement Fields", P. Anandan and R. Weiss; University of Massachusetts 186

"On Surface Reconstruction Using Sparse Depth Data", T.E. Boult and J.R. Kender; Columbia University 197

"Labelling Line Drawings of Curved Objects", J. Malik; Stanford University 209

"Solving the Depth Interpolation Problem with the Adaptive Chebyshev Acceleration Method on a Parallel Computer", D.J. Choi and J.R. Kender; Columbia University 219

"First Results on Outdoor Scene Analysis Using Range Data", M. Hebert and T. Kanade; Carnegie-Mellon University 224

"Description of Surfaces from Range Data", T.J. Fan, G. Medioni and R. Nevatia; University of Southern California 232

"Disparity Functionals and Stereo Vision", R.D. Eastman and A.M. Waxman; University of Maryland 243

TABLE OF CONTENTS

Page

SECTION III - TECHNICAL REPORTS PRESENTED (Continued)

"Evidence Combination for Vision using Likelihood Generators", D. Sher; University of Rochester 255

"Locating Cultural Regions in Aerial Imagery Using Geometric Cues", P. Fua and A.J. Hanson; SRI International 271

"The Information Fusion Problem and Rule-Based Hypotheses Applied to Complex Aggregations of Image Events", R. Belknap, E. Riseman and A. Hanson; University of Massachusetts 279

SECTION IV - TECHNICAL PAPERS NOT PRESENTED

"Probabilistic Solution of Ill-Posed Problems in Computational Vision", J. Marroquin, S. Mitter and T. Poggio: Massachusetts Institute of Technology 293

"Stereo Verification in Aerial Image Analysis", D.M. McKeown, C.A. McVay and B.D. Lucas: Carnegie-Mellon University 310

"The Terrain-Calc System", L.H. Quam; SRI International 327

"Converting Feature Values to Evidence", G. Reynolds, D. Strahman and N. Lehrer: University of Massachusetts at Amherst 331

"Binocular Image Flows", A.M. Waxman and J.H. Duncan: University of Maryland and Flow Research Company 340

"Detecting Structure in Random-Dot Patterns", R. Vistnes; Stanford University ... 350

"One-Eyed Stereo: A General Approach to Modeling 3-D Scene Geometry", T.M. Strat and M.A. Fischler: SRI International 363

TABLE OF CONTENTS

Page

SECTION IV - TECHNICAL PAPERS NOT PRESENTED (Continued)

"Stereo Correspondence: Features and Constraints", H.S. Lim and T.O. Binford; Stanford University 373

"Direct Passive Navigation: Analytical Solution for Planes", S. Negahdaripour and B.K.P. Horn; Massachusetts Institute of Technology 381

"Analysis of an Algorithm for Detection of Translational Motion", I. Pavlin, Z. Riseman and A. Hanson; University of Massachusetts 388

"Inherent Ambiguities in Recovering 3-D Motion and Structure from a Noisy Flow Field", G. Adiv; University of Massachusetts 399

"Refinement of Environmental Depth Maps Over Multiple Frames", S. Bharwani, E. Riseman and A. Hanson; University of Massachusetts ... 413

"Multiresolution Path Planning for Mobile Robots", S. Kambhampati and L.S. Davis; University of Maryland 421

"Error Detection and Correction for Stereo", R. Mohan; University of Southern California .. 433

"Geometric Grouping of Straight Lines", R. Weiss, A. Hanson and E. Riseman; University of Massachusetts 443

"On Detecting Edges", V.S. Nalwa and T.O. Binford; Stanford University 450

"Visual Surface Interpolation: A Comparison of Two Methods", T.E. Boult; Columbia University 466

"Predicting Specular Features", G. Healey and T.O. Binford; Stanford University 479

TABLE OF CONTENTS

Page

SECTION IV - TECHNICAL PAPERS NOT PRESENTED (Continued)

"A Provably Convergent Algorithm for Shape from Shading", D. Lee; AT&T Bell Laboratories 489

"Generalized Cone Descriptions from Sparse 3-D Data", K.G. Rao and R. Nevatia; University of Southern California 497

"Equivalent Descriptions of Generalized Cylinders", K.S. Roberts; Columbia University 506

"The Calibrated Imaging Lab Under Construction at CMU", S.A. Shafer; Carnegie-Mellon University 509

FOREWORD

The Sixteenth Image Understanding Workshop was held in Miami Beach, Florida on September 9-10, 1985. This workshop, attended by more than one hundred research and government personnel was the first conducted by the new Defense Advanced Research Projects Agency Program Manager for IU, LTC Robert L. Simpson, Jr. Following his welcoming remarks, LTC Simpson presented his views of the state of the DARPA research program for Image Understanding. The following paragraphs summarize LTC Simpson's views under the title, "Look Back, Look Forward."

> As the new program manager of the DARPA Image Understanding Program, I believe it is safe to say that we have arrived at an important stage in the history of this impressive research program. Looking back, we can see the evolution of image understanding and its impact on defense capabilities. Originally conceived as a five year program in 1975 by Lt Col David Carlstrom, the first several years of IU established the strong base of low-level vision techniques and knowledge-based sub-systems that began to differentiate the program from what is usually called "image processing". In the late 70's and early 80's, under the direction of Lt Col Larry Druffel, the program saw the development of model-based vision systems such as ACRONYM and the demonstration of IU techniques in more meaningful concept demonstrations such as the DARPA/DMA image understanding testbed. These demonstrations and their potential for future military use warranted the continuation of the IU program beyond its initial five year lifespan. Under CDR Ron Ohlander, IU technology continued to mature to the point that the DARPA Strategic Computing Program could justify a major application, the autonomous land vehicle.
>
> The basic theme of the IU program has remained the same as quoted by Larry Druffel in 1981:

> to investigate application of a priori knowledge to facilitate an understanding of the relationship among objects in a scene. The appropriate focus is on the world understanding....[The Image Understanding Program] is a catalyst which attempts an integration of many sciences [image processing, pattern recognition, computer science, artificial intelligence, neurophysiology, and physics] in search of methods for automatic extraction of information from imagery. (Druffel, 1981, pp. 2-3)

Looking to the future, we need to identify the real defense applications that have been made possible by the basic IU research. Otherwise, we will have difficulty justifying the continuation of IU. For those of you readers in the Department of Defense, I need to know your assessment of the value of IU to you and your mission. I solicit your evaluation and suggestions.

One major new thrust to come from the IU foundation has been the autonomous vehicle application of computer vision technology in DARPA's Strategic Computing Program. While representing an important new showcase for IU technology, we need to carefully assess the importance of basic high risk research in generic image understanding as a separate program from other DARPA initiatives such as SC.

There remains much to accomplish before we can perform visual information processing in problem solving context at the same or greater capability of human beings. This continues to be our vision for the future.

The purpose of the workshop was to present the current research results and on-going efforts to the community as a whole and to foster an interchange of technical discussions leading toward improved communications and wider utilization of mature technology. This year the workshop featured a new session consisting of invited papers by other than DARPA sponsored researchers in an effort to broaden the horizons of the group as a whole. This proceedings consists of the P.I. program reviews in Section I, the invited papers in Section II, and the technical papers presented at the workshop in Section III. At the request of the program manager, other technical papers for which time was not available for presentation are included as Section IV so that the document can serve the community as a more comprehensive reference for research papers on this research area.

This proceedings has been supplied to the Defense Technical Information Center (DTIC) and copies may be secured from that Agency by writing to the following address:

Defense Technical Information Center
Cameron Station, Bldg. #5
Alexandria, VA 22314

A small charge is assessed by the DTIC for reproduction expenses. Accession number for this proceedings is not yet available but will be assigned by the DTIC within the next thirty days. Accession numbers for previous issues are listed on the following page.

The pictures on the cover were provided by the Intelligent Systems Group, Departments of Electrical Engineering and Computer Science, University of Southern California, from one of their research efforts. Dr. Ram Nevatia, Principal Investigator for the DARPA image research at USC, described the sequences as follows:

> The pictures on the cover are from a project in making maps from aerial images at the University of Southern California (USC). This task focuses on detecting runway structures in airport images. The front cover shows part of an aerial image of the Boston Logan Airport. Detection of runways in this image is a seemingly simple task for us, but in fact has many complexities. The runways have many markings, oil and tire-tread marks and

surface material is not always homogeneous. The lower picture on the front cover shows the pieces of runways detected by the program after a sequence of processing.

The back cover shows the results at some of the intermediate steps. The first figure on the back shows the line segments detected in the image. The complexity of the task becomes much more apparent in this figure than in the original image. Next, the segments are organized in "anti-parallel" pairs of lines (lines that are parallel but of opposite contrast) based on the observation that runways, taxiways, roads etc. are characterized by such lines. The middle figure on the back cover shows the anti-parallels of a specific width and direction corresponding to the expected width and the directions of the runways. The next step is to connect the anti-parallel pairs based on a variety of evidence- first, the very strong evidence such as sharing segments and strict collinearity (giving the last figure on the back) and then much weaker evidence leading to the figure on the front cover.

We would like to point out that the pictures shown represent work in progress demonstrating the feasibility of the chosen task rather than a final, finished algorithm to perform the runway description task.

The artwork and layout designs were done by Mr. Tom Dickerson of the SAIC graphics staff. Significant assistance in handling the mailings, and in putting the proceedings together for publication was provided by Barbara Burkett, Mary Hollingsworth and Dianne Williams of the Science Applications International Corporation administrative staff. Their hard work and diligence made possible the organization of the workshop and the ability to get the proceedings papers to the printers on schedule.

Lee S. Baumann
Science Applications
International Corporation
Workshop Organizer

DEFENSE TECHNICAL INFORMATION CENTER
ACCESSION NUMBERS FOR PREVIOUS
I.U. WORKSHOPS

ISSUE	DATE	NUMBER
APRIL	1977	AD A052900
OCTOBER	1977	AD A052901
MAY	1978	AD A052902
NOVEMBER	1978	AD A064765
APRIL	1979	AD A069515
NOVEMBER	1979	AD A077568
APRIL	1980	AD A084764
APRIL	1981	AD A098261
SEPTEMBER	1982	AD A120072
JUNE	1983	AD A130251
OCTOBER	1984	AD A149496

AUTHOR INDEX

NAME	PAGE
Adiv, G.	399
Aggarwal, J.K.	89
Aggarwal, R.K.	74
Ahuja, N.	98
Aloimonos, J.	129
Anandan, P.	186
Baker, H.H.	137
Basu, A.	129
Belknap, R.	279
Bharwani, S.	413
Binford, T.O.	20, 373, 450, 479
Bolles, R.C.	137
Boult, T.E.	197, 466
Brown, C.M.	16, 129
Choi, D.J.	219
Davis, L.S.	1, 421
Duncan, J.H.	340
Eastman, R.D.	245
Fan. T.J.	232
Feldman, J.A.	16
Fischler, M.A.	40, 363
Fua, P.	271

AUTHOR INDEX (Continued)

NAME	PAGE
Hannah, M.J.	149
Hanson, A.J.	271
Hanson, A.R.	48, 279, 388, 413, 443
Healey, G.	479
Hebert, M.	224
Hoff, W.	98
Horn, B.K.P.	381
Kambhampati, S.	421
Kanade, T.	4, 163, 224
Kanatani, K.	107
Kender, J.R.	45, 197, 219
Lee, D.	469
Lehrer, N.	331
Lim, H.S.	373
Lucas, B.D.	310
Lybanon, M.	61
Malik, J.	209
Marroquin, J.	293
McKendrick, J.D.	61
McKeown, D.M.	310
McVay, C.A.	310

AUTHOR INDEX (Continued)

NAME	PAGE
Medioni, G.	117, 232
Milenkovic, V.J.	163
Mitiche, A.	89
Mitter, S.	299
Moran, R.	433
Mundy, J.L.	83
Nalwa, V.S.	176, 450
Nasr, H.N.	74
Negahdaripour, S.	381
Nevatia, R.	10, 232, 497
Panda, D.P.	74
Pauchon, E.	176
Pavlin, I.	388
Poggio, T.	25, 293
Quam, L.H.	327
Rao, K.G.	497
Rearick, T.C.	64
Reynolds, G.	331
Riseman, E.M.	48, 279, 388, 413, 443
Robertr, K.S.	506
Rosenfeld, A.	1

AUTHOR INDEX (Continued)

NAME	PAGE
Shafer, S.A.	4, 509
Sher, D.	255
Strahman, D.	331
Strat, T.M.	363
Terzopoulos, D.	156
Vistnes, R.	350
Waxman, A.M.	245, 340
Weiss, R.	186, 443
Yasumoto, Y.	117

SECTION I

PROGRAM REVIEWS BY
PRINCIPAL INVESTIGATORS

GC970145-01

IMAGE UNDERSTANDING TECHNIQUES FOR AUTONOMOUS VEHICLE NAVIGATION

Azriel Rosenfeld
Larry S. Davis

Center for Automation Research
University of Maryland
College Park, MD 20742

ABSTRACT

This report briefly summarizes research carried out during the period September 1984–October 1985 on Contract DAAK70-83-K-0018 (DARPA Order 3206). The focus of this research is on image understanding techniques applicable to autonomous land vehicle navigation. Particular emphasis has been placed on time-varying imagery analysis, but some work has also been done on stereopsis and on three-dimensional scene geometry.

1. INTRODUCTION

The Computer Vision Laboratory of the Center for Automation Research at the University of Maryland has been funded under the DARPA Image Understanding Program since 1976. The current contract, entitled "Autonomous Vehicle Navigation", was initiated in December 1982. It was funded through the U.S. Army Night Vision and Electro-Optics Laboratory in Fort Belvoir, VA under Contract DAAK70-83-K-0018, with Dr. George Jones as COTR.

The research being conducted under the contract is concerned with image understanding techniques for autonomous land vehicle navigation. (Our Laboratory is also funded under the DARPA Strategic Computing Program to develop algorithms for road network following and obstacle avoidance, to be tested on the DARPA Autonomous Land Vehicle; this work will not be described here.) Emphasis has been placed on techniques for time-varying imagery analysis, but some work has also been done on stereopsis and on three-dimensional scene geometry. Specific accomplishments in each of these areas are described in the following sections.

2. LAND VEHICLE NAVIGATION

Several studies were conducted on the project that are relevant to the problem of land vehicle navigation; they are described in the following paragraphs.

A methodology for landmark-based vehicle position determination is developed in [1]. This report describes a system by which an autonomous land vehicle might improve its estimate of its current position. This system selects visible landmarks from a database of knowledge about its environment and controls a camera's direction and focal length to obtain images of these landmarks. The landmarks are then located in the images using a modified version of the generalized Hough transform and their locations are used to triangulate to obtain the new estimate of vehicle position and position uncertainty.

Another study is concerned with simulation of road images as seen from a vehicle [2]. A model of a road is proposed which incorporates parameters that specify the curvature and slope of the road. Images synthesized using this model appear quite natural. This shows that it would be sufficient to extract these parameters in order to obtain 3D information about the road in road image analysis.

Methods of color image smoothing, segmentation and edge detection are developed in [3] in connection with extending road scene analysis algorithms color imagery. A new measure of edge information for color images based on cumulative histograms of absolute color differences is proposed. A multispectral version of the Symmetric Nearest Neighbor filter for edge-preserving smoothing and methods for image segmentation and edge detection are developed based on this measure. Experimental results show that the performance of the new algorithms is good.

Another report deals with path planning for mobile robots [6]. The problem of automatic collision-free path planning is central to mobile robot applications. In this report, we present an approach to automatic path planning based on a quadtree representation. We introduce hierarchical path searching methods, which make use of this multiresolution representation, to speed up the path planning process considerably. Finally, we discuss the applicability of this approach to mobile robot path planning. This report is included in the Proceedings of this Workshop.

3. TIME-VARYING IMAGERY ANALYSIS

The Ph.D. dissertation of Kwangyoen Wohn, under the direction of Allen M. Waxman, dealt with the analysis of image flows produced by the motions of planar

or curved surfaces. In the first part of this work, reported last year, we developed an algorithm, the Velocity Functional Method, to recover an image flow field from time-varying contours. The method follows directly from the analytic structure of the underlying image flow; no heuristics are imposed. Local image flow is modeled as a second-order Taylor series. The method computes twelve series coefficients from the normal component of image flow measured along contours. For planar surfaces in motion, the method yields the exact flow. We have demonstrated the robustness of our algorithm by carrying out the sensitivity analysis in the context of planar surfaces executing general rigid body motions in space.

The second part of Dr. Wohn's dissertation [7] explores the additional aspects of the theory for curved surfaces, where the second-order flow approximation is only locally valid. We derive the dependence of the truncation error on surface curvature and field of view. We also investigate the sensitivity of solutions to noise in the normal flow. The combined algorithms of 2-D flow estimation and 3-D structure and motion recovery are not as stable to input noise and surface structure as is the case for planar surfaces. The use of multiple frames to overcome the effects of noise is currently under study.

The Ph.D. dissertation of Muralidhara Subbarao, also under the direction of Dr. Waxman, continues the work on image flow analysis. [Dr. Waxman is now at Thinking Machines Corporation, but he continues to direct Ph.D. research at the University of Maryland.] In the first part of this research, two results on the uniqueness of image flow solutions for planar surfaces in motion have been developed [4]. The first result concerns resolving the duality of interpretations that are generally associated with the instantaneous image flow of an evolving image sequence. It is shown that the interpretation for orientation and motion of planar surfaces is unique when either two successive image flows of one planar surface patch are given or one image flow of two planar patches moving as a rigid body is given. We have proved this by deriving explicit expressions for the evolving solution of an image flow sequence with time. These expressions can be used to resolve this ambiguity of interpretation in practical problems. The second result is the proof of uniqueness for the velocity of approach which satisfies the image flow equations for planar surfaces. In addition, it is shown that this velocity can be computed as the middle root of a cubic equation. These two results together suggest a new method for solving the image flow problem for planar surfaces in motion.

Dr. Waxman and Dr. James H. Duncan have collaborated on a study of binocular image flow, as a step toward unifying stereopsis and motion analysis [5]. The analyses of visual data by stereo and motion modules have typically been treated as separate, parallel processes which both feed a common viewer-centered 2.5-D sketch of the scene. When acting separately, stereo and motion analyses are subject to certain inherent difficulties: stereo must resolve a combinatorial correspondence problem and is further complicated by the presence of occluding boundaries; motion analysis involves the solution of nonlinear equations and yields a 3-D interpretation specified up to an undetermined scale factor. A new module is described here which unifies stereo and motion analysis in a manner in which each helps to overcome the other's shortcomings. One important result is a *correlation between relative image flow (i.e., binocular difference flow) and stereo disparity*; it points to the importance of the ratio $\dot{\delta}/\delta$, rate of change of disparity δ to disparity δ, and its possible role in establishing stereo correspondence. This report is included in the Proceedings of this Workshop.

Dr. Ken-ichi Kanatani of Gunma University, Japan, who is currently at our Center, is doing research on various aspects of time-varying imagery analysis and three-dimensional vision. His first report on this project deals with the determination of 3D scene structure and motion from an image sequence without the need for point-to-point correspondence. The procedure consists of two stages: (i) determination of the *flow parameters*, which completely characterize the motion of the planar part of the object, and (ii) computation of 3D recovery from these flow parameters. The first stage is done by measuring *features* of the image sequence. The second stage is analytically expressed in terms of invariants with respect to coordinate changes. Typical features and relations to stepwise tracing are also discussed. This report is also included in the Proceedings of this Workshop.

4. THREE-DIMENSIONAL SCENE ANALYSIS

The Ph.D. dissertation of Roger D. Eastman, also under the direction of Dr. Waxman, deals with an approach to stereopsis based on the analysis of disparity fields [8]. The first part of his research investigates stereo matching constraints that derive from an analytic model of surface depth. Computational stereo is formulated as a single stage process in which potential feature point or contour matches interact to provide support for local estimates of a polynomial model of disparity (the *disparity functional*), not just estimates of disparity at isolated points. An algorithm is presented that integrates the disparity functional with multiresolution matching of zero-crossings to derive depth of surface patches. The analyticity of the disparity field is thereby exploited early in the matching process, and yields surface reconstruction as a direct byproduct of correspondence. This report is also included in the Proceedings of this Workshop.

Another stereopsis technique, based on the use of three cameras, is described in [9]. A three-camera approach for computational stereo is presented, which greatly simplifies the search problem among candidate matches and allows matching of horizontal edges. Only a simple camera geometry is considered, in which the images are rectified in the same plane. The horizontal

and vertical images are equidistant from and aligned parallel to the base image. The primitive objects of the approach are labeled edge segments, i.e., 8-connected chains of edge points with their local image properties. The matching algorithm scans through the edge segments in the base image and searches for corresponding triples of points in the three images. Local properties of points are used to classify matches. A preliminary evaluation of matches is based on goodness of match criteria. A simple postprocessing method based on contour connectivity is used to eliminate false matches. The method performs well in experiments. The basic matching algorithm generates only a few false matches and most of these can be easily eliminated.

A general discussion of projective geometry and its use in three-dimensional scene analysis is presented in [10]. Geometric properties are of key importance in the recovery of scene structure from images. It is argued that the proper formulations of the determination of scene geometry are obtained when projective geometry is used. A framework of projective geometry for computer vision is presented in brief and its applicability is demonstrated in a simple example. A computational approach to finding the necessary primitives is reviewed.

REFERENCES

1. Frederick P. Andresen and Larry S. Davis, "Visual position determination for autonomous vehicle navigation", CAR-TR-100, CS-TR-1458, November 1984.

2. Shinji Ozawa and Azriel Rosenfeld, "Synthesis of a road image as seen from a vehicle", CAR-TR-111, CS-TR-1478, March 1985.

3. Matti Pietikäinen and David Harwood, "Edge information in color images based on histograms of differences", CAR-TR-112, CS-TR-1479, March 1985.

4. Muralidhara Subbarao and Allen M. Waxman, "On the uniqueness of image flow solutions for planar surfaces in motion", CAR-TR-114, CS-TR-1485, April 1985.

5. Allen M. Waxman and James H. Duncan, "Binocular image flows: steps toward stereo-motion fusion", CAR-TR-119, CS-TR-1494, May 1985.

6. Subbarao Kambhampati and Larry S. Davis, "Multiresolution path planning for mobile robots", CAR-TR-127, CS-TR-1507, May 1985.

7. Kwangyoen Wohn and Allen M. Waxman, "Contour evolution, neighborhood deformation and local image flow: curved surfaces in motion", CAR-TR-134, CS-TR-1531, July 1985.

8. Roger D. Eastman and Allen M. Waxman, "Disparity functionals and stereo vision", CAR-TR-145, CS-TR-1547, August 1985.

9. Matti Pietikäinen and David Harwood, "Multiple-camera contour stereo", CAR-TR-151, CS-TR-1559, September 1985.

10. Ambjörn Naeve and Jan-Olof Eklundh, "On projective geometry and the recovery of 3-D structure", CAR-TR-154, CS-TR-1565, October 1985.

11. Ken-ichi Kanatani, "Structure from motion without correspondence: general principle", technical report in preparation.

GC970145-02

Image Understanding Research at CMU

Takeo Kanade

Steven Shafer

Computer Science Department

Carnegie-Mellon University

Pittsburgh PA 15213

Abstract

In the CMU Image Understanding Program we have been working on both the basic issues in understanding vision processes that deal with images and shapes, and the system issues in developing demonstrable vision systems. This report reviews our progress since the October 1984 workshop proceedings. The highlights in our Program include:

- *Victor Milenkovic has developed an edge-based trinocular (three-camera) stereo method for computing depth from images.*
- *Rick Szeliski has extended Ohta and Kanade's dynamic programming stereo method to use a coarse-to-fine multi-resolution search strategy.*
- *Ellen Walker is analyzing the object-independent geometric reasoning rules in the 3D Mosaic system.*
- *Steve Shafer is constructing the Calibrated Imaging Lab, which will provide high-precision images for stereo, motion, shape analysis, and photometric analysis.*
- *Martial Hebert has developed several algorithms for analysis of outdoor range images to extract edges, planar faces of objects, and terrain patches.*
- *Larry Matthies is analyzing motion stereo image sequences using a statistical analysis of uncertainty to yield high accuracy.*
- *Dave McKeown has started a Digital Mapping Laboratory as a local point for work in aerial photo interpretation, cartography, and computer vision. Current projects include MAPS, a large scale image/map database system, SPAM, a rule-based system for airport scene interpretation, and ARF, a system for finding and tracking roads in aerial imagery.*
- *Jon Webb is developing a high-performance vision system on a systolic machine, Warp, which will be actively used by the vision community at CMU. The Warp hardware is a reality, and almost a dozen implementation programs are now running.*
- *Gudrun Klinker has implemented the FIDO mobile robot vision and navigation system using the WARP.*
- *Chuck Thorpe, Richard Wallace, and Tony Stentz are working on the Strategic Computing Vision project, building an intelligent mobile robot for outdoor operation.*

1. Shape Understanding

1.1 Trinocular Vision

We (Victor Milenkovic) are studying trinocular stereo, a variation of edge-based stereo using three cameras instead of two [11]. Three cameras are arranged in an equilateral triangle and directed perpendicular to the plane of the triangle. Edges are extracted from the three images, and the trinocular algorithm matches corresponding edge pixels using three constraints. The first constraint is that the edge pixel positions must form an equilateral triangle. The second constraint involves the edge orientations which must satisfy a specific relation based on the perspective projection. The third constraint is photometric: we expect to find matching patterns of intensity values near the edge pixel.

Because of error, the matching algorithm uses statistical confidence measures based on the second and third constraints instead of using the constraints directly. It compares the confidence measures of competing candidate matches in order to determine which one is more likely to be correct. The algorithm also compares the confidence measure to a statistically based failure threshold in order to detect the situation where there is no match or a partial match. An example of a partial match is an edge arising from an occluding contour, in which case the edge position and orientation constraints are satisfied, but only one side of the edge satisfies the photometric constraint. A filtering method is used for resolving unorderable match candidates, in which one candidate has higher edge orientation confidence and the other has better photometric confidence.

The trinocular stereo algorithm has been applied to both real and synthetic images. It works very well on the synthetic images, and it clearly demonstrates the matching of occluding contours. The algorithm works well on real images even though the camera model is somewhat inaccurate. Basically, the algorithm works as well as the best binocular method (actually even better because it can match horizontal edges) without using any sort of continuity assumption. It does not need to assume that neighboring edge pixels have similar disparities. For each edge pixel, it searches the entire range of possible disparities, and it does not make any assumptions about the order in which the matching edge pixels appear.

A possible extension of this work is to do trinocular matching simultaneously with edge tracing. The two processes could reinforce each other: continuity along a contour would provide a means of eliminating single pixel errors of the matching scheme, while depth information would help the tracing take the correct "fork" at a T junction. Once a good set of three dimensional contours were generated, they could be reliably extended past the point at which they are visible from all three cameras. Instead of

suffering more from occlusion than two camera stereo, the trinocular algorithm would then suffer less, because more points are visible from two out of three cameras than from two out of two.

1.2 Multi-resolution Stereo Using Dynamic Programming

We (Rick Szeliski) have developed a multi-resolution version of the dynamic programming stereo algorithm introduced by Ohta and Kanade [14]. Their technique uses both intra- and inter-scanline search to obtain a disparity map starting from gray-scale real world images; we have now developed a faster version of the algorithm using a coarse-to-fine multi-resolution search strategy.

The images are first pre-processed using the DOLP Transform to build an image pyramid. The low-pass (blurred) images are used to calculate the cost function of the stereo matcher, while the band-pass images are be used to extract the edges. The stereo matching algorithm is then applied to the coarsest (smallest) image. The result of this processing (which is a list of matched edges) is then used to constrain the stereo matcher on the next finer (larger) level. To generate these constraints, it is necessary to calculate the correspondence between edges at various resolution level, but this is relatively easy and fast.

The matching proceeds until the solution for the finest level is obtained. The combined processing time for the pyramid is much lower than for single-resolution processing, since the constraints from the previous level greatly reduce the search space. In practice a speedup of 2.5 was observed. The quality of the results for the single- and multi-resolution versions were similar.

1.3 Rule-Based Geometric Reasoning for Photo Interpretation

We (Ellen Walker) have written an interactive version of the 3D Mosaic system and are using it to study the existing rules in the system and to determine new rules to be implemented. We hope to develop a system with more explicit use of domain knowledge which will be more robust, as well as capable of being easily adapted to other domains. One way we hope to do this is to add a set of low level image processing tools, along with a procedure to determine the applicability of each tool. These tools would be used for bottom-up verification of hypotheses developed by the top-down component of the system. For example, in the 3D Mosaic system, when a roof is found, edges are hypothesized from each vertex of the roof to the ground. We are studying operators which will verify these hypothesized edges to determine which ones to use under what conditions. The system is being tested on several aerial images of Washington, D.C.

1.4 The Calibrated Imaging Laboratory

We (Steve Shafer) are building a Calibrated Imaging Laboratory (CIL), which will be a facility for high-precision imaging with accurate ground truth data [12]. This laboratory will help to bridge the gap between computer vision theories, which typically depend on unrealistic assumptions about the world, and applications, which must function on real images. The CIL will span both areas by providing real images in a controlled environment, with the ability to incrementally add more complexity to the imaging situation and the scene. In all cases, accurate ground truth data will make it possible to quantitatively evaluate the performance of the methods used for image analysis. The CIL will be used to study stereo, motion analysis, geometric shape recovery, and photometric and color analysis.

The facilities of the CIL include:

- *Lighting Control* provided by a near-point light source (arc lamp) for precision shadow analysis, and a complete track lighting system for flexible general illumination.
- *Background Reflection Control* in a room with black ceiling, black carpet, and black or white curtains, with other colored backdrops as needed.
- *High-Precision Color Images* provided by a custom-built camera yielding 512x512x8 images with each pixel value being repeatable (noise-free) and linearly related to scene radiance, using color filters in a filter wheel.
- *Precision Stereo and Motion Image Sets* provided by a mobile platform with precision X-Y-Z-pan-tilt controls and a pair of CCD cameras aligned for stereo correspondence.
- *Objects* including simple objects for viewing and a scale model landscape that presents a variety of surface property, motion, and occlusion situations.
- *Calibration Data* provided by appropriate tools, including photometers, precision targets, and calibration camera filters.
- *Accurate Ground Truth Data* provided by an optical table with precision position control devices and surveyors' transits for position measurement.

2. 3D Data Analysis

2.1 Range Data Analysis for Outdoor Imagery

We (Martial Hebert) are developing algorithms for range data analysis for outdoor imagery [2]. Our goal is to develop a 3D vision system that provides a description of an unknown environment to a mobile robot. This description, a three-dimensional map of the observed scene in which regions are labeled as accessible terrain, objects, etc., will provide the necessary information for path planning and landmark recognition. We use a state of the art sensing device, the ERIM scanner, which is able to produce 64x256 range images at a frame rate of two images per second with an accuracy of 0.4 feet. This sensor combines a large field of view (30 degrees horizontal and 40 degrees vertical) and a fast acquisition rate, making it suitable for outdoor imagery analysis. For our experiments, the sensor is mounted on a testbed mobile robot so that the algorithms are tested in a realistic outdoor navigation environment.

We have developed several range data segmentation algorithms. The first goal of these algorithms is to extract three types of features: 3D edges, terrain regions divided into accessible and non-accessible ones, and obstacles divided into pseudo-planar regions. The final product of the segmentation is a graph of objects, regions, and edges. The segmentation algorithms proceeds by first extracting low-level attributes such as edge points, surface normals, surface curvature. Then, each attribute is used to derive a intermediate segmentation. Finally, the intermediate segmentations are merged together to form a consistent scene description. The complete segmentation takes about one minute on a VAX-785. This computation time will be later reduced by using the WARP systolic array processor. The segmentation programs have been used to produce input to a path planning programs for a mobile robot. Our

results show that range data analysis provides reliable information for outdoor navigation in a static environment.

The techniques described so far proceed by independently processing one image at a time. In an outdoor navigation system, consecutive images are related to each other to develop a global map. That is, the robot would grab an image every 1-10 meters such that each image is registered with respect to the previous ones. We have developed matching techniques for registering consecutive images. Matching proceeds by finding the best match between the features produced by the segmentation program; this matching in turn provides an estimate of the 3D position of the current image with respect to the current global map. The global map obtained by merging sequences of images may be used to predict the aspect of the an environment already traversed. We have tested the matching on images obtained by the testbed vehicle. Results of the matching technique are presented in these proceedings.

We are in the process of combining range data with other sources of visual information such as color images, or reflectance data, in order to obtain a more accurate environment description.

2.2 Motion Stereo Analysis

We (Larry Matthies) are studying the use of probabilistic error models in motion solving from stereo. Previous applications of stereo to motion estimation have not made full use of the available knowledge about the three-dimensional uncertainty of point features. Our simulations suggest that reductions in the estimation error on the order of a factor of ten are possible when this information is taken into account.

The basic approach to motion solving from stereo correspondences involves three steps: building a 3-D model of the scene as viewed from the first vehicle location, building another model from the second vehicle location, and finding the best fit motion that maps one model to the other. By modelling uncertainty explicitly in the first two steps, we can obtain better results for the last step of estimating motion. Previous efforts to model this uncertainty have associated a scalar with each feature point that tells the reliability of its 3D position estimate; these are then utilized as weights in a least-squares approach to motion estimation. The reliability weights are usually simply inversely proportional to the distance of each point from the vehicle.

We can do better than this by modelling the feature points as random variables with 3-D normal distributions. The means and covariances of these distributions can be estimated from the images. This leads to a a revised least squares formulation in which the weights are now 3x3 matrices formed from the covariance matrices. These effectively replace the usual distance metric by a new one that gives more weight to errors of fit in directions where positional uncertainty is low and less weight in directions where uncertainty is high. Typically triangulation leads to more uncertainty along the line of sight than perpendicular to it. Thus, when we transform one model to another, errors of fit are weighted less along the line of sight than perpendicular to it. This approach to error modelling is a standard method of photogrammetry, which we have now successfully applied to computer vision.

We have tested this algorithm in simulation by generating random sets of 3-D points, projecting them onto image planes, adding noise to the image coordinates, and completing the triangulation and motion estimation as outlined above. The system of matrix weights produces motion estimates whose standard deviations are less than those produced by the scalar method by a factor of three to ten or more, depending on the motion parameter and the distance of the points from the camera. For example, over a simulated motion of 1.0 meter straight forward, the matrix method may estimate a motion of 0.99 meters with a standard deviation of 0.01 meter, while the scalar method may estimate roughly the same motion with a standard deviation of 0.05 meters. Details of the approach and the results will appear in a forthcoming paper.

Our results suggest that error modelling can play an important part in improving the accuracy of visual ranging and motion estimation. We plan to extend this work in several directions. The first is to use it over time, as the vehicle continues to move. One approach to this is a batch least squares method that estimates all point and vehicle positions at once. A second approach is an incremental one that after each step computes only the new vehicle position and filters the new point position measurements in with the old. Both methods are likely to find use in a system that combines short-range navigation with long-range map building. Other extensions will be to detect correspondence errors and to model and estimate velocity as well as position.

3. Aerial Photo-Interpretation

We (Dave McKeown) have started a Digital Mapping Laboratory as a focal point for work in aerial photo interpretation, cartography, and computer vision.

3.1 MAPS: Large-Scale Image Map Database

One of the key issues in building systems utilizing emerging techniques in AI for applications in cartography and aerial photo interpretation is the generation and maintenance of a domain knowledge base. Loosely speaking, this "knowledge base" should contain known facts and spatial relations between objects in an area of interest, access to historical or normalcy reports, and methods to relate earth coordinates to pixel locations in digital imagery. Unfortunately, these spatial database capabilities are somewhat different than those found in traditional geographic information systems. Other issues include methods for spatial knowledge utilization and representation. For example, simply having access to cartographic descriptions does not really address the problem of how to operationalize iconic descriptions for image analysis and interpretation.

MAPS is a large-scale image/map database system for the Washington D.C. area that contains approximately 100 high resolution aerial images, a digital terrain database, and a variety of map databases from the Defense Mapping Agency (DMA). We have continued work on the MAPS image/map database system primarily in the area of integration of map data to support our work in rule-based airport scene analysis [5, 6]. Much of the geometic constraint computation in the SPAM system is derived from facilities to represent image features in geodetic coordinates <latitude,longitude,elevation>. With partial support from Defense Mapping Agency we have added approximately 50 images to our database of aerial imagery over Washington D.C. and have begun performance measurements for factual, geometric, and mixed queries on an expanded Washington D.C. map database.

3.2 SPAM: Rule-Based Intepretation of Airport Scenes

We have continued our development of SPAM, a System for Photo interpretation of Airports using MAPS [7, 8, 9]. SPAM is a rule-based image interpretation system that coordinates and controls image segmentation, segmentation analysis, and the construction of a scene model. This work uses results of the MAPS system to provide a general map description of the airport layout, and tools for spatial reasoning about size, shape, and position of various airport

features. SPAM has been run on several images of National Airport in Washington D.C..

SPAM provides several unique capabilities to bring map knowledge and collateral information to bear during all phases of the interpretation. These capabilities include:

- The use of domain-dependent spatial constraints to restrict and refine hypothesis formation during analysis.
- The use of explicit camera models that allow for the projection of map information onto the image.
- The use of image-independent metric models for shape, size, distance, absolute and relative position computation.
- The use of multiple image cues to verify ambiguous segmentations. Stereo pairs or overlapping image sequences can be used to extract information or to detect missing components of the model.

With partial support from the Defense Mapping Agency we have begun to explore other airport configurations within the SPAM system architecture. Candidate airports are Moffett Field, Dulles International, and Andrews AFB. We need to validate the current SPAM system on a variety of airports. In the process of this validation we will explore the development of tools to aid users in defining new spatial relationships and airport scene primitives. The goal of this interactive knowledge acquisition is to directly generate the rule-based component of the SPAM system from an intermediate textual representation.

3.3 Stereo Verification For Aerial Image Analysis

As part of the SPAM system, we have produced a flexible stereo verification system, STEREOSYS, and applied it to the analysis of high resolution aerial photography [10]. Stereo verification refers to the verification of hypotheses about a scene by stereo analysis of the scene. Unlike stereo interpretation, stereo verification requires only coarse indications of three-dimensional structure. In the case of aerial photography, this means coarse indications of the heights of objects above their surroundings. This requirement, together with requirements for robustness and for dense height measurements, shape the decision about the stereo system to use.

Stereo verification is used when an image region has two candidate object labels that differ in the height of the corresponding objects. Stereo is used to discriminate between the two possibilities, by selecting another image of the same land area from our image database and performing stereo analysis with the image under interpretation. To do this, we must address stereo analysis in a very unconstrained environment. Rather than simply focusing on isolated image analysis where stereo pairs are carefully controlled, we have constructed a system that can automatically perform matching and analysis using arbitrarily selected images. We have found that if knowledge-based image understanding systems are to begin to perform analysis tasks at a level of performance required for mapping and photo interpretation, they must be able to accommodate a much broader range of task uncertainty and complexity than has been previously demonstrated in any research or development system.

Stereo verification deals with a variety of problems that are not ordinarily present in isolated experiments with stereo matching and analysis. Some of these are:

- An appropriate conjugate image pair must be selected from a database of overlapping images based on criteria that would maximize the likelihood for good correspondence.
- The image pairs must be dynamically resampled such that the epipolar assumption (ie., epipolars are scan lines) used in most region-based stereo matching algorithms can be applied.
- Because the size of the areas to be matched varies greatly, the system design must be flexible and general.
- An initial coarse registration step is generally necessary because the quality of the correspondence between conjugate pairs varies greatly. In many cases the magnitude of the initial mis-registration is greater than the expected disparity shift.
- The system must analyze the stereo results and generate a symbolic description that provides an estimate of the actual height of the region in question, and the confidence of that estimate. The computation of a depth map (disparity map) is not a sufficient final result.

The results of this research indicate that image/map database issues in stereo verification influence the utility of such an approach as much as the underlying stereo matching algorithm. In fact, they are intimately related. The ability to be flexible in the selection of stereo pairs provides opportunities for multi-temporal, multi-scale, or multi-viewpoint matching. Equally as important is flexibility in the matching algorithm, especially with respect to assumptions that require nearly perfectly aligned conjugate images, a situation that is unlikely to occur in outside of the laborarory. We believe that the ability to dynamically select conjugate image pairs from a database based upon the region of interest and knowledge of the requirements of the matching algorithm is required for a fully automated aerial photo-interpretation system. Our results also indicate that stereo analysis can function as a very powerful discriminator in an image understanding system without having to perform 3D shape reconstruction. That is, coarse estimates of height, coupled with confidence in those estimates, can greatly constrain search during image interpretation.

4. Parallel Architectures for Vision

4.1 The WARP Programmable Systolic Array Processor

We (Jon Webb) are developing a high-performance vision system on a systolic machine, Warp, which will be actively used by the vision community at CMU and other sites [1, 3, 4, 19]. The Warp hardware is a reality, and almost a dozen implementation programs are now running.

We have set up three key development stages for this project:

1. Develop a critical mass of basic programs so that users (vision researchers) won't need to write routine image processing programs before they attack their own problem.

2. Develop a Standard Research Environment of which the Warp vision system is a part. The 4.2 Unix version on VME-bus based SUN workstation was selected.

3. Present demonstrations in the context of the Strategic Computing Vision project at CMU.

Webb and Kanade set out a goal of implementing 80% of the SPIDER Image Processing Packages (approximately 200 routines). So far, the following programs have been implemented in the W1

programming language, a low-level microprogramming language:

- Fast Fourier Transform (1D and 2D)
- Moravec's interest operator
- Image add, subtract, multiply, inverse, inverse square root.
- Histogram
- 3 x 3 convolution.
- DOLP (difference of low pass) transform to generate image pyramids.
- Image pyramid generation (by overlapping 4 x 4 averages)
- Edge preserving filter (routine EGPR in Spider library)
- Grayvalue translation

All of these programs run in a standard Unix environment, under Unix 4.2 BSD, running on a Sun 2/170. This Sun has been integrated into our environment at CMU.

Some of the above programs have been used to drive the Terregator, an autonomous land vehicle, at speeds of up to one kilometer per hour, which is much faster than it was capable of before Warp was used. In fact, the current speed limitation in the Terregator is not the computing, but the engine power: the Terregator cannot be driven faster than it is currently with the range of payloads it must support.

The W2 language recently became capable of compiling useful image processing programs. We have run demonstrations of the Terregator using entirely W2-generated code. Some W2 programs that have been written include:

- Arbitrary size convolution
- Median Filter
- Color segmentation of an image into different features
- Histogram
- Table lookup
- Sobel operator
- Color normalization

In the future, all Warp programming will be done in W2. We expect this to change the Warp from a machine which is fast but difficult to use into a genuine research tool.

4.2 Application of the WARP For Mobile Robot Navigation

In order to show how Warp can be integrated into complex vision systems we (Gudrun Klinker) are using Warp to perform the basic vision routines of the Fido system that was built at CMU and Stanford [18]. Fido is a mobile robot system that uses stereo vision to navigate through a room, avoiding obstacles. When running on a single processor (VAX 11/784), it spends about 80% of its time on three vision routines: data reduction, feature correlation, and the interest operator. The robot currently proceeds at a speed of 1 meter per 30 seconds.

The three vision routines have been implemented on Warp as three microprograms. Each routine uses two Warp cells. When they are run at full speed on two cells, speed improvements by factors of 3.5 (data reduction), 6 (interest operator), and 14 (feature correlation), compared to the sequential version on the VAX, can by expected. (In the final Warp system with 10 cells, the factors will be 14, 30, and 84). However, these speedups are not achieved currently, since the host system that provides Warp with data has not yet been developed completely.

Fido has been transferred from the Vax to the Warp host. The vision routines that use the Warp cells are running as independent programs. They are currently included into Fido. As a next step, Fido will be adapted to run on a faster host system, consisting of two dedicated processors for the communication with Warp and one master processor that performs the less time consuming tasks of Fido and supervises the other processors.

5. Strategic Computing Vision

Our goal for the Strategic Computing Vision project at CMU is to build an intelligent mobile robot capable of operating in the real world outdoors. We are attacking this on a number of fronts, ranging from building appropriate research vehicles to exploiting high-speed experimental computers to building software for reasoning about the perceived world [13, 15, 16, 17]. This work is summarized here because much of it is related to our image understanding research. The highlights of our progress include:

- Runs with our vehicle continuously moving along paths and sidewalks, using a television camera to sense the pavement. We have used several different image processing techniques, including color-based region classification and oriented edge tracking.
- Sonar-based runs cross-country among trees and obstacles, and at the bottom of a coal mine. Our sensors are the inexpensive Polaroid sonars. We use the overlap between the fields of view neighboring sonars, and between the field of view of the same sonar from different vehicle positions, to reason about probably empty or probably occupied regions. This gives us maps with a half foot resolution.
- Runs through the same trees using the ERIM laser scanner, using algorithms described in this proceedings.
- Successful runs using stereo vision to sense and avoid obstacles. The Fido vision and navigation system was originally built for small indoor vehicles. We have pulled out various separate modules, including path planning and stereo vision for obstacle detection.
- The first real application of the prototype Warp systolic processor. We have made several vision-based runs to demonstrate the prototype Warp. These runs showed first that vision programs were easy to put together on the Warp, and second the potential for high-speed runs when the full Warp arrives.
- Design of Navlab, a robot van. Some of the most interesting problems are in fusing data from multiple

sensors. We have designed a mobile robot, to be built in a converted van, that will have the power and size needed for these experiments. The design has room to carry stereo cameras, an ERIM scanner, sonars, several processors including a Warp, and includes room for four researchers.

- Design and first stages of implementation of a software blackboard system for connecting the output of all the sensing and reasoning programs into a single view of the world.

6. Bibliography

1. Gross, T., Kung, H.T., Lam, M. and Webb, J. Warp as a Machine for Low-level Vision. Proceedings of 1985 IEEE International Conference on Robotics and Automation, March, 1985, pp. 790-800.

2. Hebert, M. and Kanade, T. First Results on Outdoor Scene Analysis Using Range Data. Proc. DARPA Image Understanding Workshop, Miami Beach, Florida, December, 1985.

3. Kung, H.T. and Webb, J.A. Global Operations on the CMU Warp Machine. Proceedings of 1985 AIAA Computers in Aerospace V Conference, American Institute of Aeronautics and Astronautics, October, 1985.

4. Kung, H.T. and Webb, J.A. Global Operations on a Systolic Array Machine. Proceedings of IEEE International Conference on Computer Design, IEEE, 1985, pp. 165-171.

5. McKeown, D.M., Digital Cartography and Photo Interpretation from a Database Viewpoint. In *New Applications of Databases*, Gargarin, G. and Golembe, E., Eds.,Academic Press, New York, N. Y., 1984, pp. 19-42.

6. McKeown, D. M. "Knowledge-Based Aerial Photo Interpretation." *Photogrammetria, Journal of the International Society for Photogrammetry and Remote Sensing* 39 (1984), 91-123. Special Issue on Pattern Recognition

7. McKeown, D.M., Denlinger, J.L. Map-Guided Feature Extraction from Aerial Imagery. Proceedings of Second IEEE Computer Society Workshop on Computer Vision: Representation and Control, Annapolis, Maryland, May, 1984. Also available as Technical Report CMU-CS-84-117

8. McKeown, D.M. and Pane, J. F. Alignment and Connection of Fragmented Linear Features in Aerial Imagery Proceedings IEEE Computer Vision and Pattern Recognition Conference, San Francisco, California, June, 1985. Also available as Technical Report CMU-CS-85-122

9. McKeown, D.M., Harvey, W.A. and McDermott, J. "Rule Based Interpretation of Aerial Imagery." *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-7*, 5 (September 1985), 570-585.

10. McKeown, D. M., McVay, C. A., and Lucas, B. D. Stereo Verification in Aerial Image Analysis. Proc. DARPA Image Understanding Workshop, Miami Beach, Florida, December, 1985.

11. Milenkovic, V. J. and Kanade, T. Trinocular Vision Using Photometric and Edge Orientation Constraints. Proc. DARPA Image Understanding Workshop, Miami Beach, Florida, December, 1985.

12. Shafer, S. A. The Calibrated Imaging Lab Under Construction at CMU. Proc. DARPA Image Understanding Workshop, Miami Beach, Florida, December, 1985.

13. Anthony Stentz and Chuck Thorpe. An Architecture for Autonomous Vehicle Navigation. Proceedings of the Fourth International Symposium on Unmanned Untethered Submersible Technology, University of New Hampshire Marine Systems Engineering Lab, June, 1985.

14. Szeliski, R. Multi-resolution Stereo Using Dynamic Programming. Internal report of the IUS group at CMU

15. Thorpe, C., Stentz, A., and Shafer, S. An Architecture for Autonomous Vehicle Navigation. Proc. Computers in Aerospace V Conference, IEEE, Long Beach, California, October, 1985, pp. 22-27.

16. Wallace, R., A. Stentz, C. Thorpe, H. Moravec, W. Whittaker and T. Kanade. First Results in Robot Road-Following. IJCAI-85, August, 1985.

17. Wallace, R., Matsuzaki, K., Goto, Y., Webb, J., Crisman, J. and Kanade, T. Progress in Robot Road Following. submitted to 1986 IEEE Robotics and Automation Conference

18. Webb, J., Klinker, G., and Kanade, T. Parallel Vision Algorithms on a Systolic Array Machine: WARP. CMU CSD, 1985. In preparation.

19. Webb, J. and Kanade, T. Vision on a Systolic Array Machine. to appear in *Computing Structures for Image Processing*, edited by M.J.B. Duff, S. Levialdi, K. Preston and L. Uhr, Academic Press, 1986.

GC970145-03

IMAGE UNDERSTANDING RESEARCH AT USC: 1984-85[1]

R. Nevatia

Intelligent Systems Group
Departments of Electrical Engineering
and Computer Science
University of Southern California
Powell Hall Room 234
Los Angeles, CA 90089-0273

1. INTRODUCTION

Our research in this period has focused on the following topics

1. Description of 3-D surfaces and objects
2. Stereo analysis and surface interpolation - for aerial image analysis and for indoor scenes
3. Mapping from Aerial Images
4. Motion analysis
5. Parallel processing of IU algorithms
6. Conversion of software to VAX-Unix/Symbolics environments

Results of these activities are summarized below details of 1, 2 and 4 above may be found in other papers in these proceedings [1, 2, 3, 4].

2. 3-D SURFACES AND OBJECTS

We are following both surface and volume description methods. Volume descriptions have many advantages, but are not suitable for certain objects that consist of mostly thin surfaces. Also, the surface descriptions may aid in computing volume descriptions.

2.1. Surface Descriptions

Our surface description technique assumes that dense range (3-D) data is available for the visible surface. From this data, we wish to detect the lines and points in the surface that correspond to important physical properties of the surface (e.g. edges and "folds"). We propose that such descriptions can be computed from the curvature properties of a surface. In particular, we assert that the zero-crossings and the extrema of the curvature are of particular importance. While it is easy to show the correspondence between curvature properties and physical properties in principle, computation of curvature requires estimating second derivatives and hence is prone to noise or small surface variations. To handle these variations, we smooth the surface by Gaussian masks of varying variance and combine the results of different size operators. Our method has been tested on several synthetic and real images and is described in detail in [1].

2.2. Object Description

Our object description technique assumes that only sparse 3-D data of an object is available such as at object boundaries (in contrast to the dense data needed for surface descriptions). Even these boundaries may be incomplete and fragmented, as is typical of 3-D data derived from stereo, for example. In addition, the scene may contain surface markings and segments caused by noise. Our objective is to find generalized cone descriptions of the objects in the scene under these conditions. Our approach is to search for sets of boundary segments that satisfy certain expected mathematical properties expected of generalized cones. The current method is applicable only to the class of "linear, straight, homogeneous generalize cones", but we believe that it will extend to much more broader classes. For this simple case, the "contour generators" of the generalized cone are known to be coplanar and we assume the "terminator" to satisfy certain properties also. Our method and results on a few examples are given in [2].

3. STEREO ANALYSIS

Stereo analysis has been a topic of continued interest in image understanding. Earlier, we have presented a line matching stereo algorithm [5], with interesting results. We have initiated an effort to evaluate in depth the performance of various stereo algorithms, in particular those that do the correspondence matching on a line-by-line basis. In addition to providing a method for evaluation, our approach can also correct certain types of errors, as we know that disparities along a linear segment in the image must vary linearly. This approach is described in [3].

In another effort, we have been studying the problem of surface interpolation from the stereo data, as feature-based stereo gives depth information at intensity edges only. We have implemented a multi-resolution surface

[1]This research was supported by the Defense Advanced Research Projects Agency and was monitored by the Air Force Wright Aeronautical Laboratories under contract F33615-84-K-1404, Darpa order no. 3119.

reconstruction algorithm developed by Terzopoulos at MIT [6, 7]. Our implementation differs in detail, and we have tested the algorithm on a wider variety of cases than described in Terzopoulos' original work. This algorithm seems to work well when the stereo data is fairly dense, but the reconstruction quality falls when the input becomes sparse. Our experiments will be described in a USC-ISG report under preparation.

4. PARALLEL PROCESSING OF IU ALGORITHMS

The computational needs of IU algorithms are enormous and parallel processing is a promising way to speed up their performance. Parallel processing of the "low-level" IU algorithms (such as convolution and edge detection) is rather straight-forward but higher level algorithms do not have such an obvious mapping to a parallel machine. In earlier work [8], Moldovan has suggested a cellular architecture, called SNAP, that consists of an array of grid-connected processing cells with specialized communications units. In recent work, a simulator has been written for SNAP and we have examined the problems of transitioning from iconic representations to purely symbolic representations in parallel architectures. A task of object recognition assuming high-level descriptions has been analyzed. Results of these studies will be available in a USC-ISG report under preparation at this time.

5. MOTION ANALYSIS

Estimation of 3-D motion parameters and object shape from 2-D motion correspondences has been an area of active research in the IU community lately. It is possible to formulate this problem mathematically in fairly straightforward ways and to solve the resulting equations (both linear and non-linear formulations have been proposed). Unfortunately, this problem turns out to be "ill-conditioned" in the sense that small changes in input parameters cause large changes in the computed descriptions. We have studied two alternatives to alleviate this problem.

In one approach, we use the method of "regularization", popularized by Poggio [9]. In this method, a term corresponding to the smoothness of the solution is added to the error function given by the motion equations to give a more stable solution. Details of this method are given in another paper in these proceedings [4].

In another approach, we are studying the possible use of "acceleration" parameters, obtained by using 3 or more frames of motion, in addition to "velocity" parameters (obtained from 2 frames) used conventionally. Some preliminary analytical results are described in [10].

Motion analysis research has now been transitioned to our strategic computing research program because of its obvious relevance to the Autonomous Land Vehicle (ALV) program.

6. MAPPING FROM AERIAL IMAGES

Automating map-making from aerial images is a central task in our research program. In the past, we have developed methods of linear feature extraction, stereo mapping and image to map correspondence. We are now attempting to develop a system to handle complex mapping tasks; we have initially selected large commercial airports as our task domain. This domain has a variety of objects such as long, linear features (runways, taxiways, roads), a variety of buildings (hangars, terminals, etc.) and a variety of mobile objects (cars, trucks, airplanes). Further, the airport complexes are under continual changes, usually due to expansion. The mapping of this domain, thus, offers a variety of challenging problems.

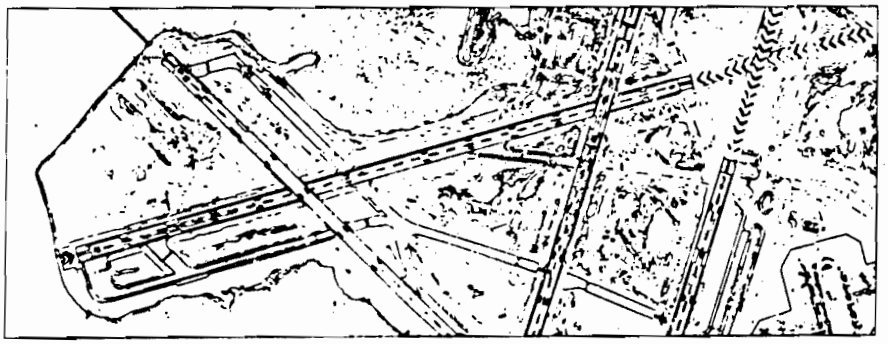
Our concentration, so far, has been in the mapping of runways and taxiways (we are pursuing mapping of buildings under separate funding from the Defense Mapping Agency). The runways and taxiways may appear to be modelled easily – namely as long, thin, rectangular strips of uniform brightness. Unfortunately, the real images are much more complex. Runways have tire and tread marks and oil spots, usually near the center and surface markings along the sides, at the two ends, and sometimes also in the middle. Also, in certain geographical locations, the runways surfaces develop defects that need to be patched: this patching is not necessarily homogeneous with the original surface material.

In the following we show steps of our analysis on an image of the Boston Logan airport. The steps are similar to those shown for the Los Angeles International Airport image in our previous report [11]. These experiments are part of our study to determine the generality of our methods and assumptions and to develop a better "knowledge-base" of this task domain.

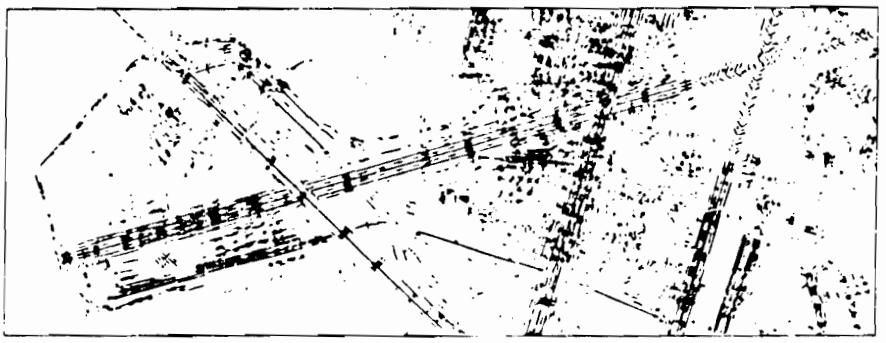
Figure 1(a) shows a 800x2200 pixel portion of the image of the Logan Airport in Boston. We first extract line segments and antiparallels (a pair of parallel line segments having opposing contrast) using our "LINEAR" software. The 8,862 segments computed are shown in figure 1(b), and the 2,212 antiparallels (apars) are shown in figure 1(c). The edges were thresholded on magnitude before line fitting, and the minimum and maximum separation between apar segments was 1 and 50 pixels respectively.

In the second step, we obtain an estimate of the direction of the runways and their width. Figure 2 shows a length-weighted histogram of the angles of the segments in figure 1(b). The histogram clearly shows the two dominant orientations, which presumably correspond to the direction of the runways in the image. We perform our processing for each orientation separately and then we combine the results.

(a): Part of Boston Logan Airport



(b): Segments detected in Fig. 1(a)



(c): Anti-parallel lines (displayed by medial axes in Fig. 1(b))

Figure 1

Figure 2: Length-weighted orientation histogram of the segments in figure 1(b)

Figure 3 shows a length-weighted width histogram of the apars in figure 1(c) that are oriented parallel to the angle estimates of runway orientation. The positive numbers on the right half of the histogram correspond to bright apars widths (the apar surounds a region brighter than its background). The negative numbers correspond to dark apar widths. Note the group of bright apars having a width of about 4 pixels. These correspond mostly to the white line markings which bound the landing surfaces. The bright and dark apars about 20 pixels wide correspond to taxiways, runway shoulders and other service roads parallel to the runways. The dark and bright apars around the 38 peak width value include the group of apars corresponding to the landing surfaces bounded by the white line markings. The bright apars come from the outside edges of the white markings and the dark apars come from the inside edges of the white line markings. We select by hand an estimate of the objects width and process each group separately.

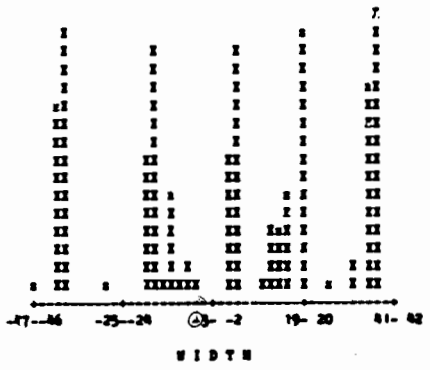


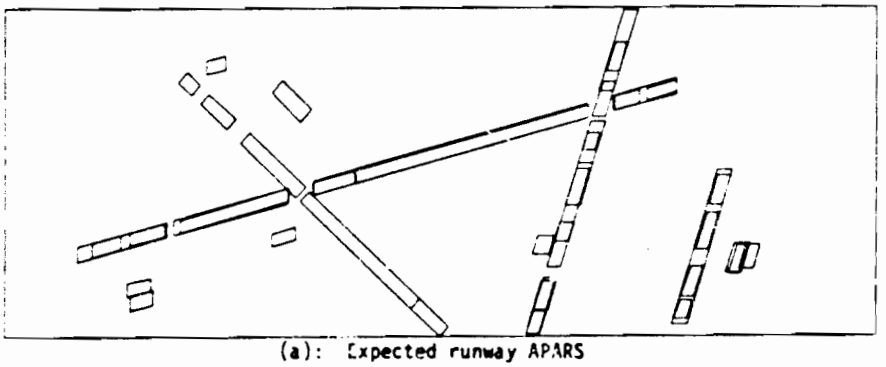
Figure 3: Length-weighted width histogram of a subset of the apars shown in figure 1(c)

In the next step we extract apars that are likely to represent portions of runways. Figure 4(a) shows the apars (as ribbons) extracted form those in figure 1(c) which are oriented in the estimated direction of the runways, and having a width between 30 and 45 pixels and an aspect minimum length-to-width aspect ratio of 2:1. The remaining steps attempt to join these portions to form longer portions.

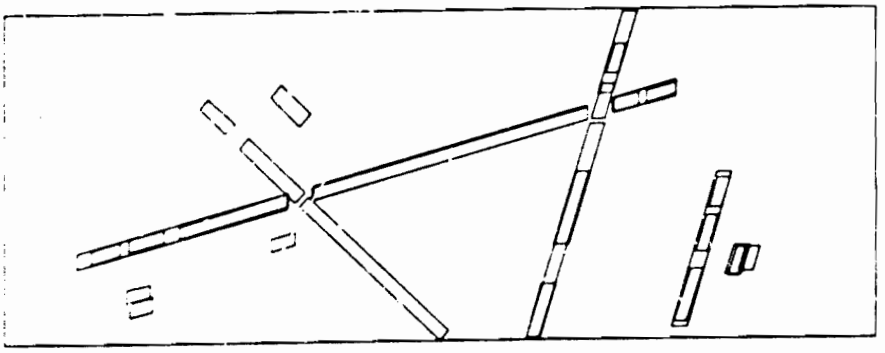
The first criteria we use to join apars is based on segment continuity. If a given segment is part of two or more apars, and the other segments in the apars are collinear, which implies that the apars have the same width and color, then the apars are joined. Figure 4(b) shows the resulting ribbons after this criteria is applied to the apars shown in figure 4(a). Note that we obtain some apars contained in wider and/or longer apars. In this example, the wider apars are bright apars formed by segments in the outside edges of the white markings, while the narrower ones are dark apars formed by segments in the inside edges of the white markings. In other cases, the narrower apars also correspond to segments in the boundaries of elongated regions on the landing surface, such as tire tread marks. In this step we eliminate those apars which are entirely contained in longer and/or wider apars. The result of this step is shown in figure 4(c).

In the next step we examine each one of the remaining apars and determine whether they are good runway candidates for further attempts to join them. If we assume that most of the line segments contained inside a ribbon must be oriented in the direction of the runway, such as those representing intermittent white markings in the center of the landing surface, as well as other markings which tend to be oriented in the direction of the runway, then we can determine which apars can be discarded by examining the line segments in a window whose size and position is that of the ribbon. Currently we require that the length-weighted orientation histogram of the segments in the ribbon show that 90% of the segments be oriented parallel to the estimated runway direction. Further refinements are needed at this step to account for common occurrences of changes in runway surface materials due to runway extensions, "patches" of surface repair having random shapes, intersections, runwey heading numbers, aircraft, and so on. For this example, no apars were discarded at this step.

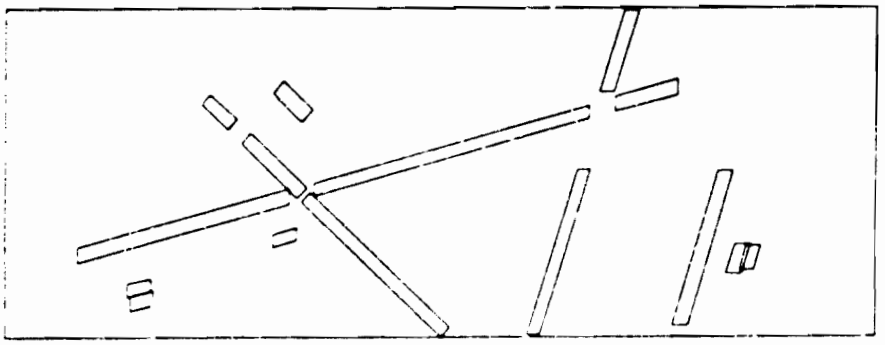
The last step in our example is similar to the one just described, except that we examine the segments in the gap between two collinear ribbons having the same width. If the length-weighted histogram of the segments in the gap shows that 90% of them are oriented in the direction of the runwey, we join the two ribbons. Figure 3(d) shows the finel result, after the short ribbons have been discarded. We note that this result provide us with a good estimate of the location of the runways in the scene, and is suitable for applying further detailed analysis leading to the description of these objects.



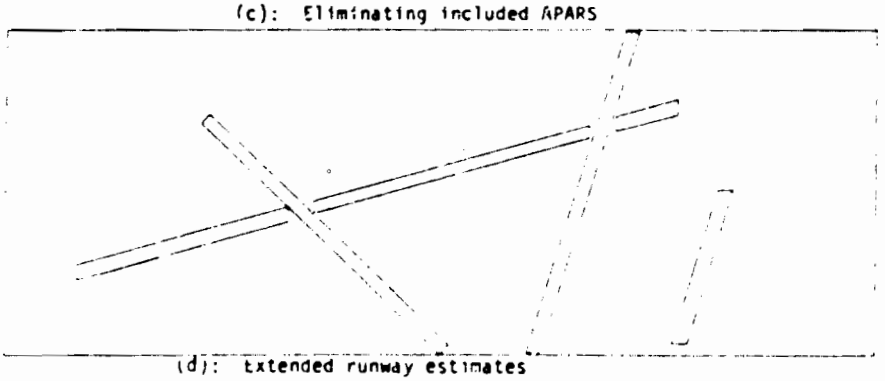
(a): Expected runway APARS



(b): Merged APARS that share segments



(c): Eliminating included APARS



(d): Extended runway estimates

Figure 4

7. SOFTWARE CONVERSION

Last, but not least, we have devoted a considerable amount of effort in the last year to converting our software. Until last year our computing engine was a DEC PDP-10 and our software was written primarily in SAIL with some small parts being in LISP (UCI-LISP). Since the beginning of 1985, we have had access to some DEC VAX 11/750 systems; some Symbolic 3640 systems have been available since summer of 1985. Our current plan is to use only these new systems as soon as is practical. Our plans are to develop most of the software in LISP, with some of the lower level programs being in C (lack of an efficient LISP for the VAXes remains a major problem). We are also attempting to maintain compatibility with "SRI IU Testbed" standards (on the VAX) and SRI ImageCalc standards on the Symbolics systems.

As our software on the PDP-10 was developed over a period of many years, the conversion effort required is rather large (and not entirely painless). Nonetheless, we have made significant progress. Our linear feature extraction system (includes "Nevatia-Babu" and "Marr-Hildreth" edge detectors, thinning, linking and linear approximation) is now available to run in C. The programs to analyze surfaces using curvature have also been converted to C on a VAX, all new development is in C. Our linear feature matching method for image to map correspondence [12], stereo matching [5] and region relaxation labelling programs [13] have been largely converted to run in LISP and are being exercised on a number of new images. Table 1 summarizes the state of this transition and some comparisons of relative run-times on different systems.

SUBSYSTEM	TARGET SYSTEM	STATUS	APPROXIMATE* PERFORMANCE
Linear feature extraction system	Unix-C	done	
Linear Convolution	Unix C	done	1
Linear Segment Finding	Unix C	done	4x
Linear Apars	Unix C	done	4x
Segment Matching	Unix - Franz Lisp	done	8x
	Symbolics Lisp	done	1/5x
Stereo Matching	Symbolics Lisp	preliminary version done	
Building Extraction	Symbolics Lisp	Basic System done	
Region Based Relaxation Matching	PDP-10 Lisp	Basic system done	7x
	Unix - Franz Lisp	Basic system done	30x
	Symbolics - Lisp	done (extended)	1
Region Segmentation	Symbolics Lisp	Color done extension started	1x

*times relative to original PDP-10

Table 1

References

1. Fan, T.J., Medioni, G. and Nevatia R., "Description of 3-D Surfaces Using Curvature," *Proceedings of DARPA Image Understanding Workshop*, Miami, Fla., December 1985.

2. Rao, G. Kashipati and Nevatia, R., "Generalized Cone Description From Sparse 3-D Data," *Proceedings of DARPA Image Understanding Workshop*, Miami, Fla., December 1985.

3. Mohan, R., "Error Detection and Correction for Stereo," *Proceedings of DARPA Image Understanding Workshop*, Miami, Fla., December 1985.

4. Yasumoto, Yoshio and Medioni, Gerard, "Robust Estimation of 3-D Motion Parameters From a Sequence of Image Frames Using Regularization," *Proceedings of DARPA Image Understanding Workshop*, Miami, Fla., December 1985.

5. Medioni, G. and Nevatia R., "Segment-Based Stereo Matching," *Computer Vision, Graphics and Image Processing*, Vol. 31, No. 1, July 1985, pp. 2-18.

6. Terzopoulos, D., *Multiresolution Computation of Visible-Surface Representations*, PhD dissertation, Massachusetts Institute of Technology, Departments of Computer Science and Electrical Engineering, January 1984.

7. Terzopoulos, D., "Computing Visible Surface Representations," *Massachusetts Institute Technology AI Lab*, No. AI Memo 800, 1985.

8. Dixit, V. and Moldovan, D.I., "Semantic Network Array Processor and Its Application to Image Understanding," *Proceedings of DARPA Image Understanding Workshop*, New Orleans, October 1984, pp. 65-71.

9. Poggio, T., "Massachusetts Institute Technology Progress in Understanding Images," *Proceedings of Image Understanding Workshop*, New Orleans, LA, October 1984, pp. 14-24.

10. Sherlat-Panahi, Hormoz, "The Motion Problem: A Decomposition-Based Solution," *Proceedings of Computer Vision and Pattern Recognition Conference*, San Francisco, Ca, June 19-23 1985.

11. Nevatia, R., "Image Understanding Research at USC: 1983-84," *Proceedings of DARPA Image Understanding Workshop*, 1984.

12. Medioni, G. and Nevatia R., "Matching Images Using Linear Features," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 6, No. 6, November 1984, pp. 675-685.

13. Price, K., "Relaxation Matching Techniques - A Comparison," *IEEE Pattern Analysis and Machine Intelligence*, Vol. 17, No. 5, September 1985, pp. 617-623.

GC970145-04

Recent Progress of the Rochester Image Understanding Project

J.A. Feldman and C.M. Brown

Computer Science Department
University of Rochester
Rochester, New York 14627

1. Robust Vision Operators

1.1. Parameter Networks and the Hough Transform

One of the most difficult problems in vision is segmentation. Recent work has shown how to calculate intrinsic images (e.g., optical flow, surface orientation, occluding contour, and disparity). These images are distinctly easier to segment than the original intensity images. Such techniques can be greatly improved by incorporating Hough methods. The Hough transform idea has been developed into a general control technique. Intrinsic image points are mapped (many to one) into 'parameter networks' [Ballard, 1983]. This theory explains segmentation in terms of highly parallel cooperative computation among intrinsic images and a set of parameter spaces at different levels of abstraction.

Earlier work on the Hough transform [Brown, 1983; Brown & Sher, 1982] has led in three directions.

1) Research toward a theory of cache accumulator arrays [Loui, 1983; Brown & Feldman, 1983]

2) Experiments with complementary HT and cache management strategies [Brown *et al.*, 1983]

3) Hardware (VLSI) designs for HT vote caches [Sher & Tevanian, 1983].

Recent efforts extend these ideas and are moving them into the parallel computing environment of the BBN Butterfly.

1.2 Bayesian Detectors

A recent extension of our work on low level operators involves the exploration of optimal operators for early vision. This is one aspect of our increased effort on formal evidence theory and its application in intelligent systems [Kyburg, 1985; Sher, 1985].

1.3 High Level Planning

In general, problem solvers cannot hope to create plans that are able to specify fully all the details of operation beforehand and must depend on run-time modification of the plan to insure correct functioning. The run-time planning idea becomes particularly important when different plan segments are being explored concurrently. These communicating segments may require sophisticated actions e.g. (do $PLAN_x$ until $PLAN_y$). These issues were studied by [Russell, 1984] in the context of a cooperative planning and execution system for manipulation tasks. A recent effort [Ballard, 1984] is examining robot planning from a task frame perspective.

2. Computing with Connections

We are continuing our interest in problem-scale parallelism, both as a model of animal brains and as a paradigm for VLSI and parallel computing [Feldman *et al.*, 1984]. Work at Rochester has concentrated on connectionist models and their application to vision. The framework is built around computational modules, the simplest of which are termed p-units. We have developed their properties and shown how they can be applied to a variety of problems [Feldman & Ballard, 1982]. We have also established powerful techniques for adaptation and change in these networks [Feldman, 1982].

A major milestone was achieved with Sabbah's thesis on massively parallel recognition of Origami-world objects [Sabbah, 1982]. Sabbah's work extended the connectionist methodology to a problem domain with several hierarchical structural levels. The resulting program is, to our knowledge, the most noise-resistant system for dealing with this level of complexity. One outcome of Sabbah's effort has been a project to build a general purpose simulator for massively parallel systems [Small et al., 1982].

The general connectionist simulator has been well tested and is being used in a number of applications. One project involved a quite detailed simulation of motor control networks of the oculo-motor system [Addanki, 1983]. Another application is to a spreading activation model of word sense disambiguation and related problems in natural language understanding [Cottrell, 1985; Cottrell & Small, 1983]. A major effort involves modelling conceptual knowledge (such as that needed for high level vision) in connectionist terms [Feldman & Shastri, 1984; Shastri & Feldman, 1984].

The second generation connectionist system has now been in active use for well over a year. Compatible versions run on the VAX systems, SUN workstations and the BBN Butterfly multi-computer. As expected, the Butterfly version makes excellent use of the available

parallelism. This is one of several approaches at Rochester to exploiting the parallel computing capabilities of the Butterfly for image understanding.

3. Parallel Computation in Image Understanding

It has been clear for many years that practical solutions to Image Understanding problems will require parallel computation. Great progress has been made in early vision and in the development of machines specialized for these computations. Intermediate and recognition level vision are more complex and it is much less obvious how to compute them in parallel. This has been a major focus of the Rochester effort for several years.

Our approach to parallelism in Image Understanding is based on the belief that general purpose (MIMD) machines will work out best for the full range of vision proglems [Feldman, 1985c]. Our work has taken a major step forward with the successful installation of a 128-processor BBN Butterfly computer. In addition to the massively parallel approaches mentioned above, we are looking at conventional vision algorithms and at general purpose parallel programming methodologies [Brown *et al.*, 1985].

4. Motion and Texture

Our interest in motion has centered around methods for extracting rigid body parameters from optic flow and intensity images. These parameters are extremely useful in navigation and target tracking. Early work showed how these nine parameters (origin, translational velocity, rotational velocity) can be extracted from flow via a Hough technique [Ballard & Kimball, 1983]. A more recent model exploits multiple channels [Bandyopadhyay, 1984]. We are also pursuing the use of these parameters to speed up the flow computations themselves [Stuth *et al.*, 1983]. A major current effort relates optical flow information to surface orientation [Aloimonos & Brown, 1984a] and sensor motion [Aloimonos & Brown, 1984b].

Recent work at Rochester has characterized the various motion algorithms according to their dependence on flow or matching and in the assumed geometric transofrm. This has lead to clarification and several new results on the number of degrees of freedom needed in different paradigms [Aloimonos & Bandyopadhyay, 1985; Bandyopadhyay & Aloimonos, 1985].

There has also been a renewed effort at exploiting texture, particularly the relative size and orientation of texture elements. This has led to nice results on shape [Aloimonos & Swain, 1985; Aloimonos & Chou, 1985] with potential applications elsewhere. Other work concerns shape from multiple views [Aloimonos *et al.*, 1985].

5. Shape

The description and recognition of complex shapes continues to be a major focus of the project. The analysis of the dot product space representation has been improved to handle certain pathological cases, and has been generalized to accommodate different criteria for the goodness of the representation.

This simple concept of shape has been applied to the problem of reconstructing three-dimensional surfaces from very sparse data. The key idea is to use appropriate shape descriptors to hypothesize a transformation which accounts for the difference in shape between successive contours. When the hypothesized transformation is minor, very simple-minded surface reconstruction techniques are sufficient. When there are major differences in shape or position between successive contours, our method hallucinates new contours, using the hypothesized shape transformation [Sloan & Hrechanyk, 1981]. A major new effort is the extraction and use of symmetries in images [Freidberg & Brown, 1984].

Hierarchical descriptions of shapes were considered in [Ballard & Sabbah, 1981] in a preliminary fashion. Our previously reported shape model [Hrechanyk & Ballard, 1982] concentrated on problems of view-invariance and attention shifting within a single prototype. This model has been extended to handle the problems of extracting primitive shape descriptions from noisy images. Our work was motivated by dissatisfactions with smoothness criteria for intrinsic image computations. Recent work extends these ideas to simple 3-D shapes [Ballard *et al.*, 1984].

The practicality of shape from shading computations and their interaction with the determination of other image parameters (such as illuminant position) was addressed by two papers in the Fall, 1982 DARPA Image Understanding Workshop. We are now applying the algorithm to real images, and want to investigate scenes with non-Lambertian reflectance functions that are unknown apriori. We want to explain how humans in fact use shading to derive shape, given the complexity of reflectance functions and imaging situations in the world. Two competing theories are that somehow the reflectance functions are derived fairly accurately by an adaptive procedure, or instead that we only 'support' a small number of reflectance functions that are selected by other cues (such as gloss).

6. General Theory of Vision

Work in our laboratory, among others, has demonstrated strong links between powerful Image Understanding techniques and computations used by animal visual systems. We have established strong ties with a wide range of visual scientists at Rochester and a variety of collaborative efforts are underway. One early project is to survey the computational similarities in natural and computer vision [Ballard & Coleman, 1983].

One part of our general effort in understanding vision (and related problems) is a comprehensive look at evidence theory in AI. One can view recognition and decision problems as combining uncertain evidence and the formal method of combination is a critical question. We have examined the traditional [Loui *et al.*, 1985] and Dempster-Shafer [Kyburg, 1985] methods and discovered some of their strengths and weaknesses. An evidential system based on the maximum-entropy principle has been quite effective in simple problems of categorization and inheritance [Shastri, 1985; Shastri & Feldman, 1985].

We have begun to exploit Rochester neurobiology expertise in order to hone and improve our connectionist modelling efforts. One difficult avenue is to specify the interface between our computational models and the state-of-the-art neurobiological picture. Our efforts in this direction are summarized in [Ballard, to appear] and the collaboration is continuing. Another effort is our attempt to develop a general framework for theories of vision that would provide a common structure for integrating studies from various disciplines [Feldman, 1985a]. These efforts are already being reflected back on our applied Image Understanding efforts [Feldman, 1985b].

References

Addanki, S., "On a Distributed Approach to Occulomotor Control," TR121, Computer Science Dept., Univ. Rochester, 1983.

Aloimonos, J., "One Eye Suffices: A Computational Model of Monocular Depth Perception," TR160, Computer Science Dept., Univ. Rochester, December 1984.

Aloimonos, J. and A. Bandyopadhyay, "Perception of Structure from Motion: Lower Bound Results," TR158, Computer Science Dept., Univ. Rochester, March 1985.

Aloimonos, J., A Bandyopadhyay and P. Chou, "On the Foundations of Trinocular Machine Vision," *Technical Digest*, Topical Meeting of Optical Society of America, Lake Tahoe, April 1985.

Aloimonos, J. and C.M. Brown, "The Relationship Between Optical Flow and Surface Orientation," *Proceedings*, 7th ICPR, Montreal, August, 1984a.

Aloimonos, J. and C.M. Brown, "Direct Processing of Curvilinear Sensor Motion From a Sequence of Perspective Images," *Proceedings*, 1984 IEEE Workshop on Computer Vision Representation and Control, Annapolis, Md., 72-77, May, 1984b.

Aloimonos, J. and P. Chou, "Detection of Surface Orientation from Texture," TR161, *Optic News*, September 1985.

Aloimonos, J. and M. Swain, "Shape from Texture," *Proceedings*, IJCAI, Los Angeles, August 1985.

Ballard, D.H., "Cortical Connections: Structure and Function," TR133, Computer Science Dept., Univ. Rochester, July 1984; to appear, *Behavioral and Brain Sciences*.

Ballard, D.H., "Task Frames in Robot Manipulation," *Proceedings*, AAAI-84, August, 1984.

Ballard, D.H., "Parameter Networks: Towards a Theory of Low-Level Vision," *Proceedings*, 7th Int'l. Joint Conf. on Artificial Intelligence, Vancouver, British Columbia, August 1981.

Ballard, D.H. and O.A. Kimball, "Rigid Body Motion from Depth and Optical Flow," *CVGIP* Special Issue on Computer Vision, 1983; also TR70, Computer Science Dept., Univ. Rochester, November 1981.

Ballard, D.H. and D. Sabbah, "Detecting Object Orientation from Surface Normals," *Proceedings*, 7th IJCAI, Vancouver, British Columbia, August 1981.

Ballard, D.H. and H. Tanaka, "Transformational Form Perception in 3D: Constraints, Algorithms, Implementation," *Proceedings*, Int'l. Joint Conf. on Artificial Intelligence, Los Angeles, CA., 964-968, August, 1985

Ballard, D.H., A. Bandyopadhyay, J. Sullins and H Tanaka, "A Connectionist Polyhedral Model of Extrapersonal Space," *Proceedings*, 1984 IEEE Conf. on Computer Vision, Annapolis, MD., May, 1984.

Bandyopadhyay, A., "Constraints on the Computation of Rigid Motion Parameters from Retinal Displacements," TR168, Computer Science Dept., Univ. Rochester, October 1985.

Bandyopadhyay, A., "A Multiple Channel Model for Perception of Optical Flow," *Proceedings*, 1984 Workshop on Computer Vision Representation and Control, Annapolis, Md., May, 1984, 78-82.

Bandyopadhyay, A., "Interest Points, Disparities and Correspondence," *Proceedings*, DARPA Image Understanding Workshop, New Orleans, La., 1984.

Bandyopadhyay, A. and J. Aloimonos, "Perception of Rigid Motion from Spatiotemporal Derivatives of Optical Flow," TR157, Computer Science Dept., Univ. Rochester, March 1985.

Bandyopadhyay, A. and J. Aloimonos, "Perception of Motion for Rigid Objects," TR169, Computer Science Dept., Univ. Rochester, forthcoming.

Brown, C.M., "Bias and Noise in Hough Transform: Theory," *Pattern Analysis and Machine Intelligence*, 1983; also, TR105, Computer Science Dept., Univ. Rochester, July 1982.

Brown, C.M., M. Curtiss, and D. Sher, "Bias & Noise in Hough Transform: Experiments," *Proceedings*, IJCAI-83, Karlsruhe, West Germany, August, 1983; also TR113, Computer Science Dept., Univ. Rochester, 1982.

Brown, C.M., C.S. Ellis, J.A. Feldman, S.A. Friedberg and T.J. LeBlanc, "Artificial Intelligence Research on the Butterfly Multiprocessor," *Proceedings*, Workshop on AI and Distributed Problem Solving, National Academy of Sciences, Washington, DC, 109-118, May, 1985.

Brown, C.M. and J.A. Feldman, "Statistical Questions Arising in the Use of Hough Techniques in Image Understanding," *Proceedings*, ONR Workshop on Statistical Image Processing and Graphics Workshop, Luray, VA., 24-27 May 1983.

Brown, C.M. and D. Sher, "Modeling the Sequential Behavior of Hough Transform Schemes," *Proceedings*, DARPA Image Understanding Workshop, November, 1982; also TR114, Computer Science Dept., Univ. Rochester, August 1982.

Cottre'l, G.W., "A Conectionist Approach to Word Sense Disambiguation." Ph.D. thesis, Computer Science Dept., Univ. Rochester, April 1985; also TR145, Computer Science Dept., Univ. Rochester, May, 1985.

Cottrell, G.W., 'Parallelism in Inheritance Hierarchies with Exceptions," *Proceedings*, Int'l. Joint Conf. on Artificial Intelligence, Los Angeles, CA, 194-202, August 1985.

Cottrell, G.W. and S.L. Small, "A Connectionist Scheme for Modelling Word Sense Disambiguation," *Cognition and Brain Theory*, 6 (1), 89-120, 1983.

Feldman, J.A., "A Connectionist Model of Visual Memory," in *Parallel Models of Associative Memory*, G.E. Hinton and J.A. Anderson (eds.), Hillsdale, NJ: Lawrence Erlbaum Associates, publishers, 1981.

Feldman, J.A., "Dynamic Connections in Neural Networks." *Biological Cybernetics*, 46, 27-39, 1982.

Feldman, J.A., "Four Frames Suffice: A Provisional Model of Vision and Space," *Behavioral and Brain Sciences*, 8, 265-289, June 1985a.

Feldman, J.A., "Connectionist Models and Parallelism in High Level Vision," *Computer Vision, Graphics and Image Processing*, 31, 178-200, 1985b.

Feldman, J.A., "Connections Massive Parallelism in Natural and Artificial Intelligence," *BYTE*, 277-284, April 1985c.

Feldman, J.A., "Energy and the Behavior of Connectionist Models," TR155, Computer Science Dept., Univ. Rochester, November, 1985d.

Feldman, J.A., "A Functional Model of Vision and Space," to appear, *Vision, Brain and Cooperative Computation*, 1986.

Feldman, J.A. and D.H. Ballard, "Connectionist Models and Their Properties," *Cognitive Science*, 6, 205-254, 1982.

Feldman, J.A. and L. Shastri, "Evidential Inference in Activation Networks," *Proceedings*, Cognitive Science Conference, Boulder, Co., July 1984.

Feldman, J.A. and L. Shastri, "Neural Nets, Routines and Semantic Networks," to appear, *Directions in Cognitive Science*, 1986.

Feldman, J.A., D.H. Ballard, C.M. Brown and S.L. Small, "Rochester Connectionist Papers, 1979-1984," TR124, Computer Science Dept., Univ. Rochester, June 1984.

Friedberg, S.A. and C.M. Brown, "Symmetry Evaluators", *Proceedings*, DARPA Image Understanding Workshop, New Orleans, LA, 1984.

Friedberg, S.A. and C.M. Brown, "Finding Axes of Skewed Symmetry," *Proceedings*, 7th ICPR, Montreal, August, 1984.

Hrechanyk, L.M. and D.H. Ballard, "Viewframes: A Connectionist Model of Form Perception," *Proceedings*, DARPA Workshop, June 1983.

Hrechanyk, L.M. and D.H. Ballard, "A Connectionist Model of Form Perception," *Proceedings*, Workshop on Computer Vision: Representation and Control, Rindge, New Hampshire, August 1982; also, *Computer Science and Engineering Research Review*, Computer Science Dept., Fall, 1982.

Kyburg, H.E. Jr., "Bayesian and Non-Bayesian Evidential Updating, TR139 (revised), Computer Science Dept., Univ. Rochester, July 1985.

Lampeter, W., "Design, Function and Performance of a System that Screens Chest Radiographs for Tumors," *Proceedings*, 1st CVPR Conference, June 1983.

Loui, R.P., J.A. Feldman and H.E. Kyburg, Jr., "Interval-Based Decisions for Reasoning Systems," *Proceedings*, AAAI Workshop on Probability and Uncertainty in AI., 1985.

Russell, D.M., "Schema-based Problem Solving," Ph.D. Dissertation, Computer Science Dept., Univ. Rochester, December 1984.

Sabbah, D., "Computing with Connections in Visual Recognition of Origami Objects," *Cognitive Science*, 9, 1, 25-50, Winter, 1985.

Shastri, L., "Evidential Reasoning in Semantic Networks - A Formal Theory and its Parallel Implementation," Ph.D. thesis, Computer Science Dept., Univ. Rochester, July 1985; also, TR166, Computer Science Dept., Univ. Rochester, September 1985.

Shastri, L. and J.A. Feldman, "Semantic Networks and Neural Nets," TR131, Computer Science Dept., Univ. Rochester, June, 1984.

Shastri, L. and J.A. Feldman, "Evidential Reasoning in Semantic Networks: A Formal Theory," *Proceedings*, IJCAI, August 1985.

Sher, D., "Template Matching on Parallel Architectures," TR156, Computer Science Dept., Univ. Rochester, July 1985.

Sher D., "Developing and Analyzing Boundary Detection Operators Using Probabilistic Models," *Proceedings*, ACM Workshop on Uncertainty and Probability in Artificial Intelligence, August, 1985.

Sher, D. and A. Tevanian, "A Hough Chip," Internal course project report, Computer Science Dept., Univ. Rochester, April 1983.

Small, S.L., L. Shastri, M.L. Brucks, S.G. Kaufman, G.W. Cottrell and S. Addanki, "ISCON: A Network Construction Aid and Simulator for Connectionist Models," TR109 Computer Science Dept., Univ. Rochester, September 1982.

Sloan, K.R., Jr. and L.M. Hrechanyk, "Surface Reconstruction from Sparse Data," *Proceedings: Pattern Recognition and Image Processing*, Dallas, Texas, August, 1981.

Stuth, B.H., D.H. Ballard and C.M. Brown, "Boundary Conditions in Multiple Intrinsic Images," *Proceedings*, IJCAI-83, Karlsruhe, West Germany, 1983.

GC970145-05

SPATIAL UNDERSTANDING

Thomas O. Binford

Stanford Artificial Intelligence Laboratory

ABSTRACT

Our goal is to build an intelligent stereo vision system which incorporates both general and special knowledge. Its purpose is stereo mapping, construction of surface maps with symbolic information, e.g. horizontal and vertical surfaces and feature analysis. A preliminary report is made of a high level stereo system which finds correspondence among extended curves, junctions, and surfaces.

The system relies on extended curves and vertices obtained from a new edge segmentation and curve linking system described in these proceedings. Another report describes psychophysics experiments to infer human mechanisms for detecting image structures by mechanisms which are parallel. Striking new results are obtained; a mechanism is suggested for computer perception. An analysis and experiments are presented concerning specular reflection in order to estimate principal curvatures of surfaces from the width of spectral peaks, and to make symbolic predictions from object models.

An algorithm is reported for labeling line drawings of opaque, curved objects bounded by piecewise smooth surfaces. The method is mathematically rigorous and complete for scenes without surface markings, and illumination discontinuities.

INTRODUCTION

The goal of this research program is to build an intelligent stereo vision system by developing comprehensive, fundamental geometric representation to express generic world constraints and special-purpose, domain-specific knowledge. This research in model-based systems is relevant to general systems as well as systems for limited classes of objects. The research is directed toward solving major problems faced in building image understanding systems:

1. The software problem: Programming major systems and application programs is the dominant computer science problem. It is accentuated by the difficulty of computational vision and the special knowledge required. We investigate automatic model-based generation of programs.

2. Computation: Vision algorithms often require about four order of magnitude more computation than VAX-class general computers provide. We develop general methods for prediction from models to cut computation by automating focus of attention, use of special-purpose shortcuts, and choice of efficient strategies for perception.

3. Segmentation: Segmentation algorithms have many weaknesses. We are building models of sensors and operators in order to automate selection of effective features tailored for individual problems. We study incorporation of available knowledge at all levels.

4. General system: The breadth of computational perception is enormous. We defis representation for generic models and constraints in order to build systems from a set of compact and uniform perceptual mechanisms with broad relevance.

One application objective of our research is construction of surface maps with symbolic information, e.g. horizontal and vertical surfaces. A second is construction of feature maps which is entirely manual and very time-consuming. A third objective measurement of surface dimensions in photointerpretation.

To meet these objectives we are building a high level stereo system intended to incorporate a broad base of constraints and knowledge at all levels of structure. This system is to incorporate inference rules and quasi-invariants for surface interpretation [Binford

31]. Stereo systems evolve to include more powerful geometric constraints. However, there is a long way to go. Most systems have used few and weak constraints, mostly restricted to epipolar correspondence.

Figure 1 shows our structure for an intelligent stereo vision system. On one hand are models, on the other hand are observations. The system makes predictions from models; it constructs observations from observations and data. The system matches observations with models. There are four gross levels of system structure:

1. object classes are functional classes in four-dimensional space-time, behavior classes;

2. surface and volume structures in three space include viewer-centered and object-centered representations;

3. image structures in two space relate image features;

4. signals and images are raw data.

Program modules construct elements in the boxes. "Shape-from-x" modules build surface observations, by direct range measurement, by stereo, by observer motion, by object motion, by shading, by photometric stereo, by inference and interpolation from image shape, or by texture. Segmentation and aggregation modules extract image features and build image structural descriptions.

Typically, models refer to models of individual objects. Here models permeate the system. There are models at all levels; at each level models range from generic to individual. Individual models have included individual aircraft and industrial parts. Generic models include generalized cylinder parts.

NEW STEREO SYSTEM

[Lim 85] reports on the status of a new stereo system. Our previous systems incorporated continuity among edge elements between epipolar lines [Arnold 78], incorporated quasi-invariants for matching corresponding edges and surface elements within epipolar lines [Arnold 80], and extended the Viterbi algorithm for making global surface correspondence within epipolar lines [Arnold 82], interpolated surfaces between edge elements within epipolar lines [Baker 81].

The new system relates extended curves, junctions, and surfaces. The system relies on high quality output from a segmentation system [Nalwa 85]. Lim has improved the determination of junctions in order to match corresponding views of junctions.

Curves must correspond if real and visible, i.e. unless extraneous in one image, missing in one image, or obscured in one image. The new system finds maximal connected components of corresponding curves; curves which are obscured are now identified by preliminary methods which are being extended. The aim is to incorporate the line labeling analysis of [Malik 85] and further geometric constraints. These conditions apply to curved surfaces. Note that quasi-invariants have already been applied in stereo vision systems [Arnold 80]. These developments make the interpretation of line drawings a key element of potentially practical systems, after it had long appeared to be an academic exercise. Likelihood of extraneous or missing curves will be modeled using a generic model of observability and by modeling segmentation operators. Some missing curves can be verified by subsequent curve verification operators. In other work, [Treindl 84] described matching strategies based on building local structures and determining the least ambiguous structures.

The system has had initial tests with one scene for which stereo line drawings were obtained automatically, and with line drawings input by hand. Even with improvements in vertex determination, some vertices in real data do not correspond. However, matching of curves is effective in these cases.

SEGMENTATION AND AGGREGATION

Segmentation describes the local structure of images in terms of features. Aggregation describes global structure of images in terms of natural relations among image features. Image structural relations often reflect spatial coherence of objects. Thus, image structures suggest candidate objects and provide a form figure-ground discrimination.

For model-based vision systems, matching models against combinatorial sets of features as ACRONYM does, is computationally expensive and limits its utility greatly. Determining candidate objects for matching cuts the combinatorics of matching, making model-based interpretation more widely relevant. This is a divide-and-conquer approach.

[Nalwa 84, 85a] describe an edge segmentation based on determining edge elements (edgels) which are discontinuities in the intensity surface. These are oriented edgels defined on a 5x5 disk by a one-dimensional step in the form of a tanh function along an edgel direction. Edgel orientation and transverse location are estimated with relatively high resolution, near information limits. Extensive error analysis was carried out on the operator.

The output of the operator is a set of disconnected edgels which know nothing about neighboring edgels. [Nalwa 85b] determine extended curves linking edgels and estimate best-fit conic curves to linked edges. They link edgels by mapping extended, directional edgels onto a high-resolution grid with half the spacing of the original grid, then thinning to preserve continuity but minimal connectivity. Chains of connected edgels are obtained by contour following between junctions.

In order to segment chains of connected edgels, dependence of tangent angle vs arc length is used. Straight segments have constant angle. Circular arcs have linear angle vs arc length. vs arc length is used to find straight line segments, to detect corners, and to choose candidate knots. Non-straight segments are described by conic sections. Arc length is a notoriously ill-defined parameter; a polygonal approximation was used to estimate arc length. A distance measure for a point from a conic was derived which offered improvements over previous formulations. This measure produces better conic fits than other programs. For conics, it is necessary to segment at inflection points. In attempting to define corners, the edgel operator is appropriate for single step edges on the disk. At corners, the operator is inappropriate, the disk limits resolution and causes averaging. A high curvature area the size of the disk is indistinguishable from a sharp corner. The program requires a minimum angle change at a corner.

Previous work has gone to determine structures beyond extended edges, especially colinear curves [Lowe 83]. This work goes on in [Vistnes 85] which describes psychophysics experiments to infer human mechanisms for detecting dotted curves in random dot backgrounds. Mechanisms were investigated which might be parallel, i.e. pre-attentive without scanning and eye motion, with less than 200 msec exposure. Interesting results were obtained:

1. grouping of dotted curves is not "diameter-limited", i.e. dotted curves are visible even for large spacing;

2. grouping of dotted curves is "background-limited", i.e. dotted curves are visible over a wide range of dot spacing and background dot spacing, as long as target dot spacing is less than background dot spacing; this scaling phenomenon is the dominant feature of these experiments;

3. results were not sensitive to line length or regularity of target dot spacing;

4. results depended on target curvature and transverse point scatter.

Because random dots formed noticeable clusters, more uniform, pseudo-random dots were generated by variations about grid positions on a coarse grid.

Results with orientation of segments indicate that contrary to a previous experiment [Riley 81], humans can discriminate textures of segments with two orientations from textures with random orientations.

INTERPRETATION OF SURFACES

[Malik 85] presents an algorithm for labeling line drawings of opaque, curved objects bounded by piecewise smooth surfaces. The scheme is mathematically rigorous and complete for scenes without surface markings, shadows, or illumination boundaries. Objects with surfaces which have point discontinuities, e.g. cones, are excluded. The analysis is for orthographic projection.

The analysis deals with generic properties for general viewpoint. The projection of neighborhoods on surfaces and edges is catalogued. Whitney's results on singularities of projection are relied upon [Whitney 55]. For a smooth surface the only singularities are folds and cusps. There are in addition, discontinuities at edges.

A result is proved that curved surfaces can be replaced by their tangent planes to give identical tangent lines at images. I.e., image vertices can be replaced by equivalent polyhedral vertices. A major objective is to achieve reasonable interpretations and avoid the large number of interpretations which even simple scenes like a tetrahedron allow. A local simplicity condition seems successful: find vertex interpretations with the minimum number of faces meeting at a vertex.

[Kirousis 84] show that the labeling problem is NP-complete. Nonetheless, by considering a reduced labeling, the practical complexity (distinct from worst case complexity) is quite low. An

efficient algorithm was determined. A program was implemented to analyze line drawings constructed by hand.

SPECULARITY

[Healey 85] analyze specular reflection for two purposes: to estimate properties of surfaces; and to make symbolic predictions from object models. They use the Torrance-Sparrow model of specular reflection from a rough surface.

Specular reflections are often the brightest, highest contrast, most prominent features in images. Thus they are often easy to find. In experiments with ACRONYM [Chelberg 84], specularities were found to be important. An effort was begun to include them in future systems. To be useful in general systems, the emphasis was on deriving symbolic, quasi-invariant results which are generic.

The model includes surface roughness by assuming a distribution of surface orientations about the nominal. It also includes a shading factor which is important at glancing incidence.

The analysis provides estimates of surface principal curvatures from the width of the specular peak. It also provides predictions of widths of spectral peaks in terms of surface principal curvatures. A set of experiments has been carried out to demonstrate these results.

REFERENCE

[Arnold 1978] D. Arnold, "Local context in matching edges for stereo vision", Proceedings Image understanding workshop, May, 1978.

[Arnold 1983] D. Arnold, "Automated stereo perception", Ph.D. thesis, Stanford University, March 1983.

[R.D.Arnold, T.O.Binford] "Geometric Constraints in Stereo Vision", Proc. SPIE Meeting, San Diego, Cal. July 1980

[Baker, H. and Binford, T.O.] "Depth from Edge and Intensity Based Stereo", Proc Int Joint Conf on AI, Aug 1981

[Binford 1981] T. O. Binford, "Inferring surfaces from images", Artificial Intelligence, Volume 17, 1981

[Chelberg 84] D. Chelberg, H.Lim, C.Cowan, "Acronym Model-Based Vision in the Intelligent Task Automation Project", Proc DARPA IU Workshop, 1984

[Healey 85] G. Healey, T.O. Binford; "Predicting Specular Features"; Proc DARPA IU Workshop, 1985.

[Kirousis 84] Kirousis, L. and C.H.Papadimitriou, "Complexity of Recognizing Polyhedral Scenes", Stanford Tech Rept 1984.

[Lim 85] H.S.Lim, T.O.Binford, "Stereo Correspondence: Features and Constraints"; Proc DARPA IU Workshop, 1985.

[Lowe 83] D. Lowe, T.O. Binford;"Perceptual Organization as a Basis for Visual Recognition"; Proc DARPA IU Workshop, May 1983.

[Malik 85] J. Malik; "Labeling Line Drawings of Curved Objects"; Proc DARPA IU Workshop, 1985.

[Nalwa 84] V.S.Nalwa; "On Detecting Edges"; Proc DARPA IU Workshop, 1984.

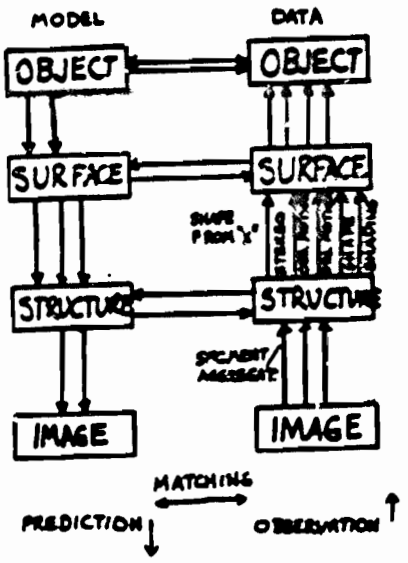
[Nalwa 85a] V.S. Nalwa, T.O.Binford; "On Detecting Edges"; accepted for IEEE PAMI, 1985.

[Nalwa 85b] V.S.Nalwa, E.Pauchon; "Edge-Aggregation and Edge-Description"; Proc DARPA IU Workshop, 1985.

[Riley 81] M.Riley; "The representation of Image Texture", MIT AI Lab Memo TR-649, 1981.

[Vistnes 85] R. Vistnes; "Detecting Structure in Random-Dot Patterns"; Proc DARPA IU Workshop, 1985.

[Whitney 55] H. Whitney; "Singularities of mappings of Euclidean Spaces; I: Mappings of the plane into the plane", Ann Math 62, 374, 1955.


MODEL
DATA
OBJECT
OBJECT
SURFACE
SURFACE
STRUCTURE
STRUCTURE
IMAGE
IMAGE
MATCHING
PREDICTION
OBSERVATION

GC970145-06

MIT PROGRESS IN UNDERSTANDING IMAGES

T. Poggio and the staff

The Artificial Intelligence Laboratory, Massachusetts Institute of Technology

Our work to date has focused primarily on the initial processes and representations of low and intermediate-level vision that decode information about the 3-D surfaces and their properties. We are now turning our efforts to the integration of different sources of information. In this report we review the work during the last year that has preceeded this new integration project and was still focused on specific problems.

In particular, we will discuss regularization theory and some of its new extensions, surface reconstruction, the computation of color, learning of regularization algorithms, parallel multigrid algorithms, representation of 3-D shape, object recognition, the development of an artificial eye-head system, the understanding of trajectories and the computation of spatial properties.

1. Early Vision

The first part of vision from images to surfaces-has been termed *early and intermediate vision*. Since at least the work of Marr (see Marr, 1982) it is widely assumed that it is mainly bottom-up, relying on general knowledge but no special high level information about the scene to be analyzed. Although this point-of-view has been embraced widely , it is important to observe that its correctness is still to be proven. In particular, it is still unclear what the nature of the 2-1/2 D sketch representation is, how different visual modules interact and their output is fused, what is the role of high-level knowledge on early visual processes. The critical problem of the organization of vision and of the control of the flow of information from the different modules and high-level knowledge is still unresolved. We are now beginning to approach this set of problems by attempting to integrate different visual modules to provide a sparse, symbolic and robust representation of space around the viewer. The main idea is that the first step in vision is fast, coarse and bottom up with the goal of providing a sparse and symbolic representation of 3-D surfaces, just sufficient for indexing a data basis and for coarse navigation. This first step must be fast and robust. For robustness and reliability under generic conditions, it exploits strongly the fusion of different low-level modules such as stereo, motion, color, edge detection and labeling of physical edges etc. The problem of interaction of the different modules is made manageable by a coarse and symbolic description of surfaces and their properties. Once coarse recognition is achieved, the mode of processing shifts to top-down: model verification, task dependent routines and representations take over, focusing on small parts of the visual field, possibly with high resolution. We believe that the rigorous analysis of individual modules of vision that we have carried out until now will play an important role in any new theory of vision.

The standard definition of computational vision is that it is inverse optics. The direct problem–the problem of classical optics or computer graphics-is to determine the images of three-dimensional objects. Computational vision is confronted with inverse problems of recovering surfaces from images. Much information is lost during the imaging process that projects a three-dimensional world into two-dimensional arrays (images). As a consequence, vision must rely on natural constraints, that is, general assumptions about the physical world to derive an ambiguous output. This is typical for many inverse problems in mathematics and physics.

As we described in the last report, the common characteristics of most early vision problems, in a sense their deep structure, can be formalized: *early vision problems are ill-posed in the sense defined by Hadamard.*

Bertero, Poggio and Torre (1986) show precisely how several problems listed in Table 1 are formally ill-posed. (See also Poggio and Torre, 1984.) The recognition that early vision problems are ill-posed suggests immediately the use of regularization methods developed

in mathematics and mathematical physics (Poggio and Torre, 1984).

1.1. Regularization theory and extensions

The main idea for "solving" ill-posed problems is to restrict the class of admissible solutions by introducing suitable *a priori* knowledge. In standard regularization methods, due mainly to Tikhonov, the regularization of the ill-posed problem of finding z from the data y :

$$Az = y$$

requires the choice of norms $\|\cdot\|$ and of a stabilizing functional $\|Pz\|$. In standard regularization theory, A is a linear operator, the norms are quadratic and P is linear. A method that can be applied is:

Find z that minimizes

$$\|Az - y\|^2 + \lambda\|Pz\|^2, \qquad (1)$$

where λ is a so-called regularization parameter.

In this method, λ controls the compromise between the degree of regularization of a solution and its closeness to the data (the first term in Equation 1). P embeds the physical constraints of the problem. It can be shown for quadratic variational principles that under mild conditions the solution space is convex and an unique solution exists.

Poggio et al. (1984, 1986) show that several problems in early vision can be "solved" by standard regularization techniques. Surface reconstruction, optical flow at each point in the image, optical flow along contours, color, stereo can be computed by using standard regularization techniques. Variational principles that are not exactly quadratic but have the same form as Equation 1 can be used for other problems in early vision. The main results of Tikhonov can, in fact, be extended to some cases in which the operators A and P are nonlinear, provided they satisfy certain conditions (Morozov, 1984).

A list of the problems that can be regularized by standard regularization theory or slightly non-linear versions of it are listed in Table 1, together with the associated regularization principle.

TABLE 1

Problem	Regularization Principle
Edge detection	$\int \left[(Sf - i)^2 + \lambda(f_{xx})^2\right]dx$
Optical Flow (area based)	$\int \left[(i_x u + i_y v + i_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)\right]dxdy$
Optical Flow (contour based)	$\int \left[(\mathbf{V}\cdot\mathbf{N} - V^N)^2 + \lambda(\frac{\partial}{\partial s}\mathbf{V})^2\right]ds$
Surface Reconstruction	$\int \left[(S\cdot f - d)^2 + \lambda(f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2)^2\right]dxdy$
Spatiotemporal approximation	$\int \left[(S\cdot f - i)^2 + \lambda(\nabla f\cdot\mathbf{V} + f_t)^2\right]dxdy$
Color	$\|I^\nu - Az\|^2 + \lambda\|Pz\|^2$
Shape from Shading	$\int \left[(E - R(f,g))^2 + \lambda(f_x{}^2 + f_y{}^2 + g_x{}^2 + g_y{}^2)\right]dxdy$
Stereo	$\int \left\{\left[\nabla^2 G * \left(L(x,y) - R(x + d(x,y), y)\right)\right]^2 + \lambda(\nabla d)^2\right\}dxdy$

1.1.1. Limitations of Standard Regularization Theory

This new theoretical framework for early vision shows clearly not only the attractions, but also the limitations that are intrinsic to the standard Tikhonov form of regularization theory. Standard regularization methods lead to satisfactory solutions of early vision problems but cannot deal effectively and directly with a few general problems such as *discontinuities* and *fusion of information from multiple modules*.

Standard regularization theory with linear A and P is equivalent to restricting the space of solution to generalized splines, whose order depends on the order of the stabilizer P. This means that in some cases the solution is too smooth, and cannot be faithful in locations where discontinuities are present. In optical flow, surface reconstruction and stereo, discontinuities are in fact not only present, but also the most critical locations for subsequent visual information processing. Standard regularization cannot easily deal with another critical problem of vision, the problem of fusing information from different early vision modules. Since the regularizing principles of the standard theory are quadratic, they lead to linear Euler-Lagrange equations. The output of different modules can therefore be combined only in a linear way. Terzopoulos (1984; see also Poggio et al., 1984, 1985) has shown how standard regularization techniques can be used in the presence of discontinuities in the case of surface interpolation. After standard regularization, locations where the solution f originates a large error in the second term of Equation 1, are identified (this needs setting a threshold for the error in smoothness). A second regularization step is then performed using the location of discontinuities as boundary conditions.

A similar method could be used for fusing information from multiple sources. a regularizing step could be performed and locations where terms of the type of the first term of Equation 1 give large errors would be identified. A decision step would then follow by setting appropriately various controlling parameters in those locations, therefore weighting in an appropriate way (for instance, vetoing some of) the various contributing processes.

In any case, one would like more comprehensive and coherent theories capable of dealing directly with the problem of discontinuities and the problem of fusing information. So the challenge for a regularization theory of early vision is to extend it beyond standard regularization methods and their most obvious non-linear versions. We will describe two such attempts: the first due to Marroquin, Mitter and Poggio (1985), is based on optimal stochastic estimation in the form proposed by Geman and Geman (1985); the second, due to Terzopoulos, attempts to apply some of the ideas of the stochastic approach to the continuous and deterministic framework of standard regularization.

1.2. A stochastic approach

In the stochastic approach, the *a priori* knowledge is represented in terms of an appropriate probability distribution whereas in standard regularization *a priori* knowledge leads to restrictions on the solution space. This distribution, together with a probabilistic description of the noise that corrupts the observations, allows one to use Bayes theory to compute the posterior distribution $P_{f|g}$, which represents the likelihod of a solution f given the observations g. In this way, we can solve the reconstruction problem by finding the estimate f which either maximises this likelihood (the so called Maximum a Posteriori or MAP estimate), or minimises the expected value (with respect to $P_{f|g}$) of an appropriate error function. The class of solutions that can be obtained in this way is much larger than in standard regularization. In particular, Marroquin, Mitter and Poggio (this volume) show under which conditions this new method leads to solutions that are of the standard regularization type.

The price to be paid for this increased flexibility is computational complexity. New parallel architectures and possibly hybrid computers of the digital-analog type promise however to deal effectively with the computational requirements of the methods proposed here. We will discuss in a later section these new parallel architectures.

The main results obtained by Marroquin (1985) are (see Marroquin et al., this volume):

(A) The connection between error criteria and the optional Bayesian estimates. The MAP estimate is not optimal for the most natural error criteria. Optimal estimators can use Metropolis-type algorithms, but do not need annealing schemes.

(B) They derive algorithms for optimal estimation of surface reconstruction and for signal matching (for instance, for binocular stereo).

(C) New hybrid parallel computers (digital and analog) on which coupled MRF models map naturally.

1.2.1. Learning

A critical problem for the practical feasibility of the MRF approach to vision is the estimation of the parameters of the MRF and noise models that are required in each specific case. Estimation from the noisy data alone is in general a very difficult problem although specific results have been obtained (see section 4.1). It may be however of some interest to consider an easier problem, that of *supervised learning* with noiseless examples, in the context of MRF formulations.

1.3. Controlled-continuity constraints and discontinuity detection

Terzopoulos (1986b) proposes a class of controlled-continuity constraints for the regularization solution of ill-posed visual problems in an arbitrary number of dimensions. These generic constraints may surpass conventional smoothness constraints by offering precise control over spatial continuity. This may make it possible to optimally reconstruct, by solving well-posed variational principles, not only continuous regions, but multiple-order discontinuities as well (global smoothness constraints are clearly inadequate for reconstructing discontinuities).

Controlled-continuity constraints are constructed from generalized spline kernels, which characterize regions, combined with continuity control functions, which characterize boundaries. Unlike global smoothness constraints, which lead to quadratic variational principles, controlled-continuity constraints are formulated as nonquadratic functionals, in general. The resulting variational principles characterize nonlinear, spatially noninvariant PDEs. Under certain conditions these problems have Bayesian estimation and nonlinear filtering interpretations (this formulation was in fact suggested by the Bayes approach of Geman and Geman (1984) and Marroquin (1984, 1985a).

Terzopoulos (1986b) observes that optimal estimation of discontinuities with controlled-continuity constraints poses a nonlinear problem in distributed parameter identification. The parameters to be identified from incomplete data are the continuity control functions. This too is an ill-posed inverse problem and it is most naturally regularized, in recursive fashion, using embedded controlled-continuity models of lower dimensionality.

A particular controlled-continuity model, the thin plate surface under tension, has been applied in a framework for computing visible-surface representations from multiple sources of impoverished visual information, and a multistage, nonlinear optimization strategy has been developed for identifying and reconstructing discontinuities (Terzopoulos, 1985a). The strategy is deterministic, and efficient.

1.4. Regularizing Color

Another problem that is presently approached with regularization techniques is the computation of spectral reflectance of surfaces. An important goal of computer vision is to recover the invariant spectral reflectance properties of an object's surface independently of the illuminant. Poggio and Hurlbert (unpublished) are developing a class of regularization algorithms for "solving" this problem, that we now describe in some detail (since it will remain otherwise unpublished).

The data measured by the sensors are:

$$S^{\nu}(x) = \int a^{\nu}(\lambda) R(\lambda, x) E(\lambda, x) d\lambda, \qquad (2)$$

where ν labels the spectral type of sensors ($\nu = 1, ...3$ for R-G-B), $a^{\nu}(\lambda)$ is the spectral sensitivity of the νth-type sensor (or filter) and $S^{\nu}(x)$ its output. $R(\lambda, x)$ is the *surface reflectance* and $E(\lambda, x)$ the *effective illumination* intensity that takes into account 3-D shading effects and possibly shadows (Hurlbert, 1985).

Equation (2) shows that R and E cannot be determined from S and a uniquely without additional constraints. In any case, there is clearly a factor - the relative scaling of E and R - that cannot be determined unless the illuminant is observed directly.

Regularization algorithms may exploit several constraints that are usually true for the albedo and the reflectance. The main constraints listed by Hurlbert and Poggio are (see also Rubin and Richards, 1984):

1) Typical illuminants and typical albedos have a finite (and small) number of degrees of freedom. In other words, they can be described as linear combinations of a fixed set of basis functions (notice that in different situations - such as different lighting conditions - different basis may be required and that the illuminant basis is different, in general, from the albedo basis). This assumption - called *the spectral regularization assumption* has been stated by several authors (for a recent review see Maloney, 1985).

2) Typically, $E(x)$, the "effective illumination", changes with x more slowly than $R(x, \lambda)$ (apart from cases when surface changes abruptly or sharp shadows are present); $R(x, \lambda)$ is either constant or changes

sharply (at material edges). This is the *spatial regularization assumption*, essentially due to E. Land (see also Horn, 1974)

3) Usually, the effective illumination has the same dependence on x at each λ:

$$E(x,\lambda) = E(x)K(\lambda) \qquad (3)$$

This is the *single source assumption* because Equation (3) is satisfied when there is a single, homogeneous light source, even in the presence of shading. Shadows may be a problem in some cases, because of self-illumination effects (see Rubin and Richards, 1984).

Two additional assumptions that we are not using directly but may be easily incorporated in our framework are:

4) In many scenes, it may be possible to assume that the average surface reflectance in each wavelength band is 'grey'.

5) For many naturally occurring reflectances (neglecting specular reflections and highlights) the ratio of the largest value to the smallest is no larger than about 10 (Lettvin).

Assumption 1 allows to rewrite Equation 2 as

$$S^\nu(x) = T^\nu_{i,j} e^i(x) r^j(x) \qquad (4)$$

where the tensor T is defined as

$$T^\nu_{i,j} = \int d\lambda a^\nu(\lambda) p^i(\lambda) q^j(\lambda)$$

where $\nu = 1,...4$ and $i,j = 1,...N$, the p's and the q's are the basis functions for the illuminant and for the albedo, respectively, and summation over repeated indices is tacitly assumed.

Assumption 2 means that $e^i(x)$ will change slowly (apart from surface discontinuities) and $r^j(x)$ will be either constant or change sharply.

Assumption 3 means that $e^i(x) = e^i\varsigma(x)$.

Thus the ratio between the intensity $S^\nu(x)$ measured in one spectral channel and the intensity measured in another should change in a noiseless situation only at albedo boundaries, (where the albedo changes), and should be invariant for changes in the effective illumination.

Poggio and Hurlbert are using the probabilistic tools introduced by Geman and Geman (1.45) and developed further by Marroquin (see Marroquin et al., this volume) to develop algorithms that can exploit these constraints. The basic idea is to define Markov Random Fields associated to albedo and effective illumination with a probabilistic structure reflecting the relevant assumption outlined earlier.

A (vector) MRF can be defined, corresponding to the illuminant components and taking continuous values $\vec{e}(x)$ on the lattice. Another MRF corresponds to the albedo $\vec{r}(x)$ and is also continuous. The a priori probability distributions are Gibbs distributions with potentials $U(e)$ and $U(r)$. We choose these to satisfy Assumption 2. For instance

$$U_1(e) = U(\vec{e}_i,\vec{e}_j) = (\vec{e}_i - \vec{e}_j)^2 \quad \text{for} |i-j| = 1$$
$$= 0 \quad \text{otherwise}$$
$$U_2(r) = U(\vec{r}_i,\vec{r}_j) = \beta((\vec{r}_i - \vec{r}_j)^2) \quad \text{for} |i-j| = 1$$
$$= 0 \quad \text{otherwise}$$

where $\beta(z)$ is a symmetric function that is 0 for $z = 0$, is large for small values of $|z|$, and may go to 0 again for large values of $|z|$ (a standard Ising model may suffice). In this way slow changes of $\vec{r}$ are penalised, whereas either constraint values or sharp changes carry no cost.

For the observations we assume that $\vec{S}(x)$ correspond to samples of the image affected by additive gaussian noise giving a potential term in the *a posteriori* distribution of the type

$$U_0 = \frac{1}{2\beta^2}\sum_i (\vec{S}_i - \vec{T}_{\mu\lambda} c^\mu_i r^\lambda_i)^2$$

We can exploit assumption (3) by defining two additional MRF, one corresponding to the albedo discontinuities and the other to effective illumination discontinuities (due for instance to discontinuities in the surface or in its normal).

For this we consider the boundary indicator $D(x) = \frac{I^1(x)}{I^1(x)+I^2(x)}$ (we assume here for simplicity $\nu = 1,2$). Then we define the (observable) segmentation index d_i as $d_i = T[D(x)]$, where

$$T(x) = 0 \quad \text{if } |x| < \text{threshold}$$
$$1 \quad \text{otherwise}$$

We can also define an (observable) b_i as $b_i = T[d/dx\sum I^\nu(x)]$. We then define the *intensity boundary index* $c_i = b_i(1 - d_i)$, which marks sharp changes of in-

tensity likely to correspond to sharp changes in effective illumination but not changes in albedo (the threshold is set on the basis of noise estimates).

We are now ready to introduce our third MRF – the *segmentation process* – that should correspond to albedo boundaries. Its observation model provides a potential

$$U_3(d,s) = \alpha \sum (1 - \delta(s_i - d_i))$$

where δ is the delta function.

A similar potential describes our fourth MRF – the *line process* (marking effective illumination boundaries)

$$U_4(d,l) = \alpha \sum (1 - \delta(l_i - c_i))$$

We now couple all these MRF's together. The line process is coupled to the effective illumination by substituting $U_1(e)$ with

$$U_1(e,l) = \frac{1}{T}(e_i - e_j)^2(1 - l_{ij}) \quad \text{for} |i-j| = 1$$
$$= 0 \quad \text{otherwise}$$

The segmentation process is coupled to the albedo process by

$$U_2(r,s) = \frac{1}{T}\beta(r_i - r_j)^2(1 - s_{ij}) \quad \text{for} |i-j| = 1$$
$$= 0 \quad \text{otherwise}$$

where $\beta(z) = 0$ if $(r_i - r_j) = 0$ and otherwise is large (Ising model). We may also add a cost associated with the fact that both effective illumination edges and albedo boundaries are continuous contours and there cannot be too many of them (see Marroquin, 1985a, p. 129, Fig. 14). The total potential corresponding to the a posteriori Gibbs distribution is then

$$U_p(\vec{e},\vec{r},l,I,s) =$$
$$= U_0 + U_1 + U_2 + U_3 + U_4$$

As a performance criterion we will use a mixed criterion, where $\vec{e}$ and $\vec{r}$ should be as close as possible to their true values and we should make as few errors as possible in the assertion about the presence or absence of boundaries l and s. Thus the optimal estimates are the *posterior mean* for $e_i^\nu(x)$ and $r_i^\nu(x)$ and the *maximizer of the a posteriori marginals* for l_i and s_i (Marroquin et al., this volume).

We can compute these estimates by using Montecarlo methods of the type discussed by Marroquin (1985a,p. 132, 1985; see also Marroquin et al., this volume).

Under some special conditions, the optimal estimation of r,l reduces to the standard quadratic variational principle, of the type of equation (1), developed by Hurlbert and Poggio (see Poggio, Torre and Koch, 1985; Hurlbert, 1985) for the computation of color. First of all, we assume that there is no line or segmentation processes; furthermore, the p and q are chosen so that $T_{ij}^\nu = T_{ij}^\nu \delta_{ij}$, implying that U_0 has the form $U_0 = \sum |i^\nu - (r^{\tilde{\nu}} + e^{\tilde{\nu}})|^2$. Finally, the potential U is assumed to have the form (for continuous x)

$$\left\| \frac{d}{dx} e^\nu(x) \right\|^2 + \left\| \frac{d^2}{dx^2} G * r^\nu(x) + \gamma \frac{d^2}{dx^2} r^\nu(x) \right\|^2$$

where G is a gaussian filter with standard deviation σ.

If we choose as performance criterion a functional that penalizes very much any single error, the optimal extimate of r and e turns out to be the MAP estimate. This corresponds in turn to r and e that minimises the quadratic potential function $U(r,e)$. The problem is then of the standard regularization type. Minimisation of U is equivalent to filtering $I^\nu(x)$ (assuming continuous data) through linear filters to obtain R^ν and E^ν. For instance,

$$R^\nu(\omega) = \frac{\lambda\omega^2}{1+\lambda\omega^2} \cdot \frac{1}{(1+\lambda\omega^\nu e^{-\omega^2\sigma^2} + \lambda\omega^4)(1+\lambda\omega^2) + 1} I^\nu(\omega)$$

. Although the last standard regularization algorithm is computationally simple, this is not so for the full stochastic model that consists of no less than four MRFs! It is likely however that the full MRF model could be used to refine a rough solution found (easily) in terms of the intensity boundary and the segmentation indeces (and possibly the standard regularization filtering).

1.5. Color computation without regularization

A non-regularization algorithm for computing spectral reflectances has been studied by Yuille (1984). A similar theory was developed earlier by and independently Wandell and Maloney (1981; see Maloney, 1985). The

method for recovering the surface reflectance of an object when the incident illumination is unknown is based on having enough different spectal type of sensors. The theory assumes that, for most objects viewed under normal lighting conditions, the illumination and surface reflectance can be expanded in terms of a finite number of basis functions (see Equation 4). If we restrict ourselves to Land's Mondrian world, which consists of flat rectangular patches of different colors, each patch will yield a number of non-linear equations for the illumination and reflectances of each patch. As the boundary between two adjacent patches is arbitrarily thin, the illumination will be the same on either side of it and it is possible to show that, given enough types of photoreceptors in the eye, the equations for adjacent patches can be combined to solve for the illumination (which does not need to be constant over the whole scene) and the reflectance functions of the two patches up to an overall scaling factor. If there are three basis functions for the illumination and reflectance then four types of sensors are needed. The method may be extended to deal with general objects by defining an edge-detection-like operation which detects boundaries between regions of different color and which then determines the color as in the Mondrian case. The color thus determined on the boundaries of objects can then be propagated inwards.

1.6. Learning a regularization algorithm

If we want to develop a powerful and flexible vision system we need ideally to endow it with the capability of improving its algorithms from experience and learning new ones from examples. It is in fact possible to learn some standard regularizing algorithms from examples without having to formulate explicitely the variational principle that embeds the relevant physical constraints (Hurlbert and Poggio, 1984). This is especially important because the exact form of the relevant constraints depend on the specific type of situation. For instance, the regularizing constraints on the spectral properties of illumination for solving the problem of computing surface reflectances is somewhat different for outdoors vs. indoor situations. The capability to learn the exact form of the "regularizer" algorithm from examples could provide an efficient way of approaching this problem.

Minimization of the regularization principle Equation 1 corresponds to a *regularizing* operator, i.e. a system, acting on the input data y and providing, as an output, the regularized solution z. We want to show that this regularizing operator can be synthetized by associative learning from a set of examples. Our argument consists of two simple claims. The first claim is that the regularizing operator corresponding to quadratic variational principles is linear. The second one is that any linear mapping between two vector input spaces can be synthetized by an associative scheme based on the computation of the pseudoinverse of the data. We explain now in more detail our argument.

(a) Variational principles of the form of expressions (1) provide a regularized solution as a linear transformation of the data. The reason for this is that the Euler-Lagrange equations are linear partial differential equations and therefore define the solution z as a linear functional of the data y (depending on the boundary conditions). For simplicity, we will restrict ourselves in this paper to the discrete case of Equation 1, in which $\vec{z}$ and $\vec{y}$ are vectors and A and the Tikhonov stabilizer P are matrices and A does not depend on the data. Equation 1 becomes then

$$\|A\vec{z} - \vec{y}\|^2 + \lambda\|P\vec{z}\|^2, \tag{5}$$

where $\|\cdot\|$ is a norm . The minimum of this functional will occur at its unique stationary point $\vec{z}$. Setting to zero the gradient of the functional of Equation 5 gives the minimum vector $\vec{z}$ as the solution of the systems of linear equations

$$(A^T A + \lambda P^T P)\vec{z} = A^T\vec{y} \tag{6}$$

It follows that the solution $\vec{z}$ can be written as

$$\vec{z} = S\vec{y} \tag{7}$$

and is therefore a linear transformation on the data vector $\vec{y}$. It is important to notice that the linear operator (when it is not space invariant, i.e. a convolution operator) may depend in general on the given lattice of data points.

(b) Imagine now that a set of noiseless input vectors $\vec{y}$ are available together with the corresponding regularized solutions $\vec{z}$. Arrange these vectors in two matrices Y and Z. The problem of synthetizing the regularizing operator S that provides the regularized solution $\vec{z}$ for each vector $\vec{y}$ is then equivalent to " solving" the following equation

$$Z = SY \tag{8}$$

and finding the matrix S. A general solution to this problem, that is optimal in the least-squares sense, is provided by

$$S = ZY^{+} \qquad (9)$$

where Y^{+} is the pseudoinverse of Y. This is the solution which is most robust against errors, if equation 9 admits several solutions and it is the optimal solution in the least-squares sense, if no exact solution of equation 9 exists. It is of particular interest for practical applications that the pseudoinverse can be computed in an adaptive way by updating it when new data become available. Practically the numerical computation of the pseudoinverse may be ill-conditioned and is therefore advisable to regularize it by using Tikhonov's method (Tikhonov and Arsenin, 1977). We had good results in the "regularized" implementation of the learning of the color algorithm.

Equation 9 shows that the standard regularising operator S (parametrised by the lattice of data points) can be synthetized without need of an explicit variational principle, if a sufficient set of correct (in the regularization sense) input output data pairs is available to the system.

We plan to study the extension of this linear learning scheme to nonquadratic regularization principles (of the type described earlier). An obvious scheme simply involves finding the nonlinear operator that minimizes an appropriate "distance" between the data and the solution set.

1.7. Parallel algorithms: Concurrent multigrid coordination

Terzopoulos (1983, 1986a) has shown that multilevel relaxation methods are an effective tool for designing highly efficent optimization algorithms for early vision. Algorithms of this type have been developed for computing useful multiscale regularization solutions to problems in image analysis and in the computation of 3D surfaces from images. The algorithms are amenable to implementation on fine-grained, massively parallel hardware, such as the Connection Machine.

In multilevel algorithms, the primary relaxation operations on each of the levels must be coordinated consistent with optimizing the given objective functional. Originally, we employed a standard, recursive multilevel coordination strategy which proves to be very effective on sequential computers. However, the recursive strategy activates only a single level at any given time. Hence, it makes rather poor use of available processors in a highly parallel implementation.

Terzopoulos (1985b) has developed a new multilevel coordination strategy that exploits a greater degree of parallelism. In contrast to the recursive coordination schemes, the new coordination strategy is fully concurrent; it maintains processors on all the levels busy performing simultaneous relaxation operations.

The concurrent coordination strategy then aims to optimize a multilevel energy functional consisting of the sum of three terms: (1) a summation of the discrete form of the given functional on each level of a multigrid hierarchy, (2) a summation of functionals coupling each level (except the finest) to the next finer level, and (3) a summation of functionals coupling each level (except the coarsest) to the next coarser level. The interlevel coupling functionals are designed so that the scheme will be convergent. Each involves a parameter. The coupling parameters are modified during the iterative process such that there is an initially strong but gradually weakening coarse-to-fine interaction, which accelerates convergence, and an initially weak but gradually strengthening fine-to-coarse interaction, which yields consistent accuracy on all levels.

In addition to making full use of all available processing elements, the concurrent strategy is significantly easier to implement than the recursive schemes, not only on parallel, but even on conventional computers.

1.8. Parallel hardware for regularization

Our discussion suggests a classification of vision algorithms that maps naturally into parallel digital computer architectures that are now under development. Standard regularization, when sufficient, leads to two classes of parallel algorithms. Algorithms for finding minima of a convex functional such as steepest descent or the more efficient multigrid algorithms developed for vision (see previous section) can always be used. They can be replaced by convolution algorithms if the data are given on a regular grid and the operator A in Equation 3 is space-invariant. In the later case, the regularized solution is obtained by convolving the data through a precomputed filter. The MRF approach leads to algorithms either of the Metropolis type or specific for the problem (Marroquin, 1985). All these algorithms may be implemented by parallel architectures of many processors with only local connections and by hybrid computer architectures (Marroquin et al., this volume, Poggio, Torre and Koch, 1985; Koch, Marroquin and Yuille, 1985).

2. Shape representation

Brady and his associates continued to develop a new representation of two-dimensional shapes called *smoothed local symmetries* (Brady and Asada, 1984). This work, described in previous reports, represents both the bounding contour of a shape fragment and the region that it subtends or encloses. It involves contsructing a representation, the *curvature primal sketch* (Asada and Brady, 1984), of the significant changes in curvature along the contours of the shape. The work described here extends the representation to a larger class of shapes, including surfaces, and shows how to generate complex semantic network descriptions of objects, which can then be learnt. The representation has been successfully used in applications in vision and robotics. In work reported elsewhere Anita Flynn (Flynn, 1985) has successfully adapted the curvature primal sketch description for robot navigation involving multiple sensors.

2.1. Local Rotational Symmetries

Smoothed Local Symmetries provide stable and perceptually appropriate representations only for regions that are more or less elongated. Fleck (1985) has developed a companion representation, called Local Rotational Symmetries, for round regions, including irregularly circular or oval regions, round bumps and ends of elongated regions, hexagons, spirals, and round regions broken up by attachment or occlusion. Like some implementations of Smoothed Local Symmetries, her implementation of Local Rotational Symmetries computes representations of grey-scale image at multiple resolutions. A new feature of this implementation is that it allows the regions found at one resolution to guide analysis of the image at finer resolutions. Thus, exhaustive computation can be done only locally at each resolution, which makes the computation more efficient and suppresses computation of certain symmetries which are not perceptually salient. Further, to create representations at multiple resolutions, this implementation smooths the grey-scale image before extracting region boundaries, rather than smoothing region boundaries. This type of smoothing proves to be more robust on real input images. A companion re-implementation of Smoothed Local Symmetries is being developed.

2.2. Towards a Surface Primal Sketch

Ponce and Brady (Ponce and Brady, 1985) continued previous investigation of surface descriptions based on concepts of differential geometry (Brady, Ponce, Yuille, and Asada, 1985). They implented the *surface primal sketch*; a representation of significant surface changes in dense depth maps analogous to Marr's (Marr, 1976) *primal sketch* representation of image intensity changes. The implemented program detects, localizes and symbollically describes: (1)*steps*, where the surface height function is discontinuous, (2) *roofs*, where the surface normal is discontinuous, (3) *smooth joins*, where a principal curvature is discontinuous and (4) *shoulders* consisting of two roofs and a step viewed obliquely. The program performs well on range maps, generated by laser data, of objects of varying complexity.

2.3. Learning Shape Descriptions

Brady and Connell have recently combined the work on the Smoothed Local Symmetries shape representation (Brady and Asada 1984) with a modified form of Winston's learning system (Winston, 1981, Winston, 1982). The resulting program generates complex semantic network descriptions of objects directly from images (Connell and Brady, 1985a, Connell and Brady, 1985b, Connell, 1985). Furthermore, from a series of examples the system produces a model of the objects it has seen by generalizing their descriptions. These models are then used to recognize other members of the class. Using this system, the relation between an object's form and its function has been briefly explored (Brady, Agre, Braunegg, and Connell, 1984). An important focus of the shape representation work has been to determine methods for segmenting an imaged object into parts which can be described in the semantic network formalism. This decompositon is guided by the principles of smooth continuation and compactness of regions. Another focus has been to identify various types of structural approximation and ways to detect them. As suggested by Marr (Marr and Nishihara 1978) such a hierarchical structuring is useful in matching objects to class models. Four forms of structural abstraction have been found to be particularly useful and have been incorporated into the learning and matching algorithms used by the system.

3. Object recognition

Grimson and Lozano-Perez have continued their work on object recognition from sparse, noise sensory data.

Previously, we reported on a technique for recognizing occluded objects, which assumes polyhedral models of the objects of interest, and simple measurements of the position and surface orientation of small patches of surface. The technique searchs for consistent matchings between the faces of the object models and the sensory measurements, using simple geometric constraints in a standard backtracking tree search.

In the past year we have extended our work in several ways. First, we have performed theoretical analysis on the expected combinatorial efficiency of the technique, and have shown that the results of extensive simulations of the process are consistent with expected theoretical bounds. Second, we have tested the technique on several different types of real data, including sonar, laser, tactile and visual data.

Third, we have investigated alternate types on simple geometric constraints. In particular, the original technique used a set of decoupled constraints, by considering independently (1) distances between faces, (2) angles between face normals, and (3) components of vectors between faces in the direction of the face normals. While these are simple constraints to implement, and are remarkably effective at reducing the space of possible solutions, they do not completely solve the problem, since they only apply to pairs of faces, and not to the interpretation as a whole. We have consider coupled constraints of the same form as an alternative. Here, rather than simply testing whether the assignment of two data points to a pair of object faces is consistent with the measurements, we actually compute the range of possible positions along the face that the point could take. These ranges are propagated as additional points are added to the interpretation, so that each new constraint tends to reduce the range of possible positions. This continues until either there is not feasible range of positions, or until all the data are accounted for. Experiments with these coupled constraints indicates that while the portion of the search space which must be explored is reduced, the additional computational cost of performing that search tends to outweight the advantages of the reduced search.

Fourth, we have investigated additional strategies for reducing the amount of search required to find the interpretation. In particular, we have added configuration hashing techniques as a method for rapidly selecting small portions of the search space that are likely to lead to consistent interpretations. These techniques apply to both two dimensional and three-dimensional problems, and significantly improve the performance of the algorithm, without loss of accuracy.

Fifth, we have investigated techniques for automatically selecting additional places for obtaining sensory data. Since we are using only sparse, noisy data, it is frequently the case that more than one interpretation is consistent with that data. To completely solve the recognition and localization problem, we need to acquire additional sensory data, until only one interpretation is feasible. While random acquisition processes will eventually converge to a unique interpretation, we have also considered techniques that will optimally select additional sensing positions for disambiguating multiple interpretations. These techniques have been implemented and sucessfully tested.

4. Towards visuo-motor coordination

4.1. An eye-head system

A robot in a complex visual environment, where objects and the robot are allowed to move relative to each other in three dimensions, may require a sophisticated system for relocating the lines of sight of its binocular vision system. The requirements for such a system may be not unlike those for the primate oculomotor system: (1) locate and fixate objects of interest,

(2) stabilize the images of such objects despite object or self movement, and (3) reduce the bandwidth of visual information by the use of a small, high-density photoreceptor array which requires sequential relocation to different spots in a wide field of view. The ability to relocate the lines of sight may, by offering several related views, provide simplifying constraints on visual computations that are otherwise ambiguous when performed on a single static image. To study these issues we have designed and built and eye-head sstem. The system will be the input to our "vision machine", which we are now developing. In particular, it will allow hilg-level processes to direct gaze and attention to specific parts of the 3-D scene.

The MIT eye-head robot consists of a platform upon which are mounted two Hitachi solid state cameras (250 by 320 pixel array) and four rotatable mirrors, two for each camera. The platform has two axes of rotation (shown in Figure 1). Stepping motors act along these axes to change the pan and pitch angles of the platform on which the cameras are mounted. In front of each camera are two mirrors, one immediately in front, called the inner mirror and one to the side, called the outer mirror. Each mirror may be deflected by a galvanometer upon which it is mounted. Thus, the mirrors

of one camera provide "vertical" and "horizontal" deflection of its line of sight which is independent of the pan and pitch angles of the motors, and of the mirrors in front of the other camera. The motors and the mirrors provide redundant degrees of freedom for tracking in 3-D space. The motors are limited to moving both cameras at once, so the mirrors must be used at least to verge the lines of sight to the same point in space.

Using 25 mm lenses, the area of any one image corresponds to approximately 17 deg vertically (the direction that is associated with deflections of the inner mirror) and 17 deg horizontally (the direction that is associated with deflections of the outer mirror). Each line of sight may be deflected over a vertical range of roughly 18 deg by the inner mirror, and a horizontal range of 18 deg by the outer mirror. The ability to rotate the entire platform using the stepping motors greatly extends the range of angles which the cameras can view. The motors can rotate the platform through pitch angles of ± 70 deg from level and through pan angles of ± 180 deg.

To move the eye-head robot six devices must be controlled: four galvanometers and two stepping motors. A central computer, the LISP machine, accesses the image input via a frame grabber board and employs a digital control algorithm to generate mirror position commands and motor speed commands (Figure 2) An A/D converter changes the mirror commands into analog voltages in the range which the four galvanometer controllers accept as input for determining the positions of the mirrors. The LISP machine sends the stepping motor commands via a parallel digital bus to an Intel 8031 microprocessor-based controller. The microprocessor decodes the 8-bit command and handles the I/O intensive task of generating timed step sequences for the motors.

K. Cornog, T. Poggio and K. Nishihara implemented control algorithms for the guidance of "eye" and "head" movements during the tasks of fixating and tracking an object in 3-D. Our algorithms, executed in LISP, allow the eye-head robot to track an object moving up to 15deg/sec and to hold the image stable to within 3 pixels. K. Cornog has reviewed control strategies of the primate oculomotor system. She has has compared the control and performance of the robot's binocular fixation and tracking systems, to those of the smooth pursuit, saccadic and eye-head coordination systems of the primate. They find, for instance, that a control system including both positional and velocity feedback improve the ability of the robot to fixate and stabilize the image of a moving target.

4.2. Catching a ball

We are very adept at using the purely two-dimensional information we get from our retinae to manoever and react to the three-dimensional world: witness the tennis player returning a 100 m.p.h. serve. How we manage to reconstruct the three-dimensional character of the world from these two-dimensional representations has been a lively subject of research in the last ten or fifteen years. One principle that has emerged unifying many of these ideas is the need for constraints to allow the visual system to interpret the images it receives as three-dimensional. These constraints come from assumptions about the nature of the situation that produced the image.

Saxberg and Poggio (Saxberg, 1985) have looked at how gravity can be used as a constraint in the case of a free fall trajectory projected onto an image plane by central projection. We showed that in principle there is enough information in the time dependent projected trajectory to exactly reconstruct the original parabolic trajectory in three-dimensions.

We examined several methods for deriving the initial conditions of the trajectory from the trajectory even in the presence of noise. Two techniques turned out to function quite well on simulated trajectories in the presence of noise: one uses a filter whose width varies with time to filter out noise from the trajectory data as it accumulates; the other uses a least-squares technique to solve directly for the initial condition parameters from the accumulating noisy projected trajectory. In both cases, good estimates of the initial conditions were achieved from simulated projected trajectory data even in the presence of considerable noise. Performance depended on the amount of noise added, the sampling rate, and the duration of the trajectory.

Saxberg also ran a limited test of the two methods on image data. In one case, he simulated the image data of a high contrast ball traveling in a parabola; in another, he used an actual video tape of a tennis ball. We applied a very simple thresholding and chord drawing technique to identify the center of the ball in each image, and used this projected trajectory information as the input to the two routines described above. With the synthesized image data, both techniques gave estimates of initial conditions that were within 5-10 per cent of the true initial conditions; with the video-tape and the thrown ball, where initial conditions were not easily measured, the two techniqes gave plausible estimates which agreed very closely with each other.

5. Visual Routines

Our work on visual routines addresses the second major stage of visual information processing - the application of the information in the early representations to object recognition, visually-guided manipulation and more abstract visual thinking. A fundamental requirement on the processing at this stage is the robust and efficient computation of an open-ended variety of shape properties and spatial relations. Humans surpass by far the ability of current automated methods of spatial analysis to cope with visual problems such as "Is there a dot inside a closed curve?", "Is the figure labelled A above and to the right of the one labelled B?", or "Which of the figures are colinear?" Both in terms of speed and the class of inputs handled, the performance difference is huge. The limitations of current approaches to spatial analysis seem to stem from a lack of appropriate representations and algorithms, as well as a need for specialized parallel architectures.

To address the analysis of spatial relations, Ullman (1984) proposed the notion of visual routines - sequences of elemental spatial operations, drawn from a small, fixed set, which are applied to the early visual representations. The set of elemental operations should be such that, by combining them in different ways, routines for an open-ended set of abstract spatial relations and properties can be defined. These elemental operations must be powerful, robust, and very efficient.

Ullman's proposal raises four major questions for research. First, what set of basic operations will support the spatial analysis computations which are needed in the course of recognition, visual problem-solving, etc. Second, how are these elemental operatins integrated into routines for establishing specific relations, and what are the general principles governing this integration. Third, by what means can visual routines be selected and controlled - for example, in the course of processing a scene, what triggers them, and how is their order of execution determined. Finally, how can visual routines be assembled or modified to meet new requirements.

Mahoney's work so far has focussed on the first two problems. Ullman (1984) made some specific suggestions for basic operations, including shift of the processing focus, indexing "odd-man-out" locations, boundary tracing, area coloring, and location marking. Working from these, we detailed possible visual routines for a range of visual tasks which were posed in the context of schematic drawings. These tasks are related, for example, to the interpretation of terrain maps, and, in certain cases, the interpretation of edge images derived from real scenes. Building on a very simple implementation of the preceeding basic operations, Mahoney has tested a number of routines proposed for solving problems like "count the dots that are inside curves", "find a curve nesting two or more other curves", "find a location that is not inside any curve", and some simple figure/ground separation tasks. This line of work is mainly aimed at exposing issues pertaining to the integration of the proposed basic operations into useful visual routines, and providing further and more detailed requirements on the set of basic operations. A longer term goal is to implement what could be thought of as a "programming system" for spatial analysis, applicable in a variety of practical contexts. The system would provide a set of basic operations and general purpose visual routines. The visual routines for a particular application would build on these. We plan demonstrate this idea for the domain of simple terrain maps in particular. A system of this type could be used "stand-alone", or it might be integrated into a larger vision machine.

These experiments also highlight the need for novel approaches, at the level of representation and algorithm, to providing operations such as boundary tracing or area coloring. It is easy to generate examples which would present difficulties to the straightforward implementations of these operations. The main thrust of our research is to invent algorithms and supporting representations for very fast and general boundary tracing. It is common for boundaries to fragmented, superimposed upon background figures, or comprised of very abstract curvilinear structures, such as texture changes or proximity groupings of small figures. A general boundary tracing operation must cope well with all of these cases, without suffering a substantial sacrifice in processing rate. Similar considerations also apply to the area coloring operation.

We are also exploring local, parallel methods of detecting blobs - areas in the input that are significantly different from their surroundings - along with a measure of how conspicuous they are. The goal of this processing is to enable initial analysis to be applied selectively to these interesting areas, and, sometimes, to make the boundaries of these regions explicit for later input to tracing operations. For example, in the processing of a complex scene, it would often be useful for the routines which initiate recognition to be directed first to areas which correspond to the larger or otherwise more prominent objects, rather than to the finer details.

Related to this, Koch and Ullman (1984) have addressed the problem of how simple networks can account for selective shifts in visual attention. They proposed, as one of the early representations, a pointwise representation of saliency in various local properties such as color, orientation, direction of movement, disparity, etc.

A selective mapping exists between this saliency representation and a central representation, such that at any given time the latter contains the properties of only a single location in the visual field. The main selection criterion is saliency, and Koch and Ullman proposed implementation by a winner-take-all network built on a specific hierarchical, pyramid-like architecture most of whose connections are local, and whose processing elements perform only simple operations such as addition or multiplication, and do not process symbolic information such as addresses. They also suggested additional selection rules which can account for similarity and proximity effects, and changes in the selected location in time.

READING LIST

Asada, Haruo and Michael Brady. "The Curvature primal sketch", Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 758, 1984.

Brady, Michael, and Haruo Asada. "Smoothed local symmetries and their implementation", *Int. J. of Robotics Research*, **3 (3)**, 1984.

Brady, Michael, Philip Agre, David Braunegg, and Jonathan Connell. "The Mechanic's Mate", **ECAI 84: Advances in Artificial Intelligence**, T. O'Shea (ed.), Elsevier Science Publishers B.V., North-Holland, Amsterdam, 1984.

Brady, Michael, Jean Ponce, Alan Yuille, and Haruo Asada. "Describing surfaces", *Computer Vision, Graphics, and Image Processing*, 1985.

Brady, Michael. "Representing shape," *Proc. IEEE Conference on Robotics*. Atlanta, 1984.

Brady, Michael, and Alan Yuille. "An extremum principle for shape from contour," *IEEE Trans. Pattern Analysis* **6(3)**, 1984.

Brady, Michael, and Alan Yuille. "Representing three-dimensional shape," Romansy Conference, Udine, Italy. 1984.

Brooks,M.J. and Horn, Berthold K.P. "Shape and Source from Shading", A.I. Memo 720 (1985)

Buchsbaum, G. "A spatial processor model for object color perception," *J. Franklin Inst.*, 310, 1980.

Canny, John F. "Finding edges and lines," Massachusetts Institute of Technology Technical Report 720, 1983.

Connell, Jonathan H., and Michael Brady. "Generating and generalizing models of visual objects," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 823, 1985.

Connell, Jonathan H., and Michael Brady. "Learning Shape Descriptions," *IJCAI 85 Proceedings*, 1985.

Cornog, Katherine H. "Smooth Pursuit and Fixation for Robot Vision," Massachusetts Institute of Technology Department of Electrical Engineering and Computer Science Master's Thesis, 1985.

Fleck, Margaret. "Local Rotational Symmetries," Massachusetts Institute of Technology Department of Electrical Engineering and Computer Science Master's Thesis. 1985.

Flynn, Anita M. "Redundant sensors for mobile robot navigation," Massachusetts Institute of Technology Department of Electrical Engineering and Computer Science Master's Thesis, 1985.

Geman, Stuart, and Don Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, **6**, 1984.

Grimson, W.E.L. **From Images to surfaces**, Massachusetts Institute of Technology Press, Cambridge, Mass., 1981.

Grimson, W.E.L. "Surface consistency constraints in vision," *Computer Vision, Graphics, and Image Processing*, **24**, 1983.

Grimson, W. E. L. "A computational theory of visual surface interpolation," *Phil. Trans. R. Soc. London B*, 1982.

Grimson, W. E. L., and T. Lozano-Pérez. "Model-Based Recognition and Localization from Sparse Range or Tactile Data," *International Journal of Robotics Research*, **3** 1984.

Grimson, W. E. L., and T. Lozano-Pérez. "Model-based Recognition and Localization from Tactile Data," IEEE Computer Society Int. Conf. on Robotics, Atlanta, March 1984.

Grimson, W. E. L., and T. Lozano-Pérez. "Recognition and Localization of Overlapping Parts from Sparse Data in Two and Three Dimensions," IEEE Computer Society Int. Conf. on Robotics, St. Louis, March 1985.

Grimson, W. E. L., and T. Lozano-Pérez. "Model-Based Recognition and Localization From Sparse Range Data," in **Techniques for 3-D Machine Perception**, A. Rosenfeld (ed), North Holland, Amsterdam, 1985.

Grimson, W. E. L. and T. Lozano-Pérez. "Search and Sensing Strategies for Recognition and Localiza-

tion of Two and Three Dimensional Objects," Third Int. Symp. on Robotics Research, Gouvieux, France, October 1985. Published by MIT Press, Cambridge, Mass.

Grimson, W. E. L., and T. Lozano-Pérez. "Recognition and Localization of Overlapping Parts from Sparse Data," in Three-Dimensional Vision Systems, T. Kanade (ed), Kluwer Academic Publishers, 1985.

Grimson, W. E. L. "The Combinatorics of Local Constraints in Model-Based Recognition and Localisation from Sparse Data, Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 763, 1984.

Grimson, W. E. L., and T. Lozano-Pérez. "Recognition and Localization of Overlapping Parts from Sparse Data," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 841, 1985.

Hillis, D. "The Connection Machine," Massachusetts Institute of Technology Department of Electrical Engineering and Computer Science Ph.D. Thesis, 1985.

Hopfield, J.J. "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Natl. Acad. Sci. USA*, 81, 1984.

Horn, Berthold K.P. "On Lightness," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 295, 1974.

Horn, Berthold K.P. Robot Vision, MIT Press and McGraw-Hill (1985).

Horn, B.K.P., "Obtaining shape from shading information", in: *The Psychology of Computer Vision*, P.H. Winston, ed., McGraw-Hill Publ., New York, 115-155, 1975.

Horn, B.K.P., "Understanding image intensities", *Artificial Intelligence*, 8, 201-231, 1977.

Horn, B.K.P., and Schunck, B.G., "Determining optical flow", *Artificial Intelligence*, 17, 185-203, 1981.

Hurlbert, Anne. "Color computation in the visual system," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 814, 1985, in press.

Hurlbert, Anne, and Tomaso Poggio. "Associative learning of standard regularizing operators in early vision," Massachusetts Institute of Technology Artificial Intelligence Laboratory Working Paper 264, 1984.

Kirkpatrick, S., Gelatt, C.D., Jr., and Vecchi, M.P. "Optimization by simulated annealing," *Science*, 220, 1983.

Koch, Christof, and Ullman, Shimon. "Selecting one among the many: a simple network implementing shifts in selective visual attention", Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 770, C.B.I.P. Paper 003, 1984.

Koch, Christof, Jose Marroquin, and Alan L. Yuille. "Analog 'neuronal' networks in early vision," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 751, 1985.

Land, Edwin H. "Recent advances in retinex theory and some implications for cortical computations: colour vision and the natural image," *Proceedings of the National Academy of Sciences*, 80, 1983.

T. Lozano-Pérez and W. E. L. Grimson, "Recognition and localisation of overlapping parts from sparse data," Second Int. Symp. on Robotics Research, Kyoto, Japan, August 1984. Published by MIT Press, Cambridge, Mass.

Maloney, Laurence T. "Computational approaches to color constancy" Stanford University Tech. Report 1985-01, 1985.

Marr, David."Early processing of visual information," *Phil. Trans. R. Soc. London* B275, 1976.

Marr, David, and Keith Nishihara. "Representation and recognition of the spatial organisation of three dimensional shapes," *Proc.R.Soc.Lond. B.* 200, 1978.

Marroquin, J. 1984 "Surface reconstruction preserving discontinuities," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 792, 1984.

Marroquin, J. "Optimal bayesian estimators for image segmentation and surface reconstruction," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 839, 1985.

Marroquin, J. "Probabilistic solution of inverse problems," Ph.D. Thesis. Massachusetts Institute of Technology , 1985.

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. "Equation of State Calculations by Fast Computing Machines." *J. Phys. Chem.* 21, 1953.

Morozov, V.A. Methods for Solving Incorrectly Posed Problems, Springer-Verlag, New York, 1984.

Poggio, Tomaso. "Early vision: from computational structure to algorithms and parallel hardware" *Computer Vision, Graphics, and Image Processing*. 31, 1985.

Poggio, Tomaso, and Vincent Torre. "Ill-posed problems and regularization analysis in early vision," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 773, C.B.I.P. Paper 001, 1984.

Poggio, Tomaso, and Christof Koch. "An analog model of computation for the ill-posed problems of early vision," Massachusetts Institute of Technology Artificial Intelligence Laboratory 783, C.B.I.P. Paper 002, 1984.

Poggio, Tomaso, Vincent Torre, and Christof Koch. "Computational vision and regularization theory," *Nature* 317, 1985.

Poggio, Tomaso, Harry Voorhees, and Alan L. Yuille. "Regularizing Edge Detection," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 776, 1984.

Ponce, Jean, and Michael Brady, "Towards a surface primal sketch." Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 824, 1985.

Rubin, John, and Whitman Richards. "Colour Vision: Representing Material Categories," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 764, 1984.

Saxberg, Bror V. H. "Parameters of a Three-Dimensional Free-Fall Trajectory From its Two-Dimensional Central Projection ," Massachusetts Institute of Technology Department of Electrical Engineering and Computer Science Master's Thesis, 1985.

Terzopoulos, Demetri. "Multilevel computational processes for visible surface representation," *Computer Vision, Graphics, and Image Processing* 24, 1983.

Terzopoulos, Demetri. "Multiresolution Computation of Visible Surface Representations," Massachusetts Institute of Technology Department of Electrical Engineering and Computer Science Ph.D. Thesis, 1984.

Terzopoulos, Demetri. "Computing visible-surface representations," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 800, 1985.

Terzopoulos, Demetri. "Concurrent multilevel relaxation algorithms," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 851, 1985.

Terzopoulos, Demetri. "Image analysis using multigrid relaxation methods," IEEE Trans. Pattern Analysis and Machine Intelligence, 1986, in press.

Terzopoulos, Demetri. "Regularization of inverse visual problems involving discontinuities," IEEE Trans. Pattern Analysis and Machine Intelligence, 1986, in press.

Tikhonov, A. N. and V. Y. Arsenin. Solution of Ill-Posed Problems, Winston and Wiley Publishers, Washington D.C. 1977.

Torre, Vincent, and Tomaso Poggio. "On edge detection," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 768, 1984.

Ullman, Shimon. "Visual routines," *Cognition*, 18, 1984.

Wandell, Brian, and Laurence Maloney. "Computational methods for colour identification," *Proceedings Optical Society of America*, 1984.

Winston, Patrick H. "Learning new principles from precedents and exercises," *Artificial Intelligence* 19, 1981.

Winston, Patrick H. "Learning by augmenting rules and accumulating censors," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 678, 1982.

Yuille, Alan L. and Tomaso Poggio. "Scaling Theorems for Zero-crossings," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 722, 1983.

Yuille, Alan L. and Tomaso Poggio. "Fingerprints Theorems for Zero-crossings," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 730, 1983.

Yuille, Alan L. "A method for computing spectral reflectance," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 752, 1984.

GC970148-07

THE SRI IMAGE UNDERSTANDING RESEARCH PROGRAM

Martin A. Fischler (Principal Investigator)

Artificial Intelligence Center
SRI International, Menlo Park, California

ABSTRACT

The SRI Image Understanding program is a broad effort spanning the entire range of machine vision research. Its three major concerns are: (1) to develop an understanding of the physics and mathematics of the vision process, (2) to develop a knowledge-based framework for integrating and reasoning about sensed (imaged) data, and (3) to develop a machine based environment for effective experimentation, demonstration, and evaluation of our theoretical results, as well as providing a vehicle for technology transfer. This report describes recent progress in all three areas. In particular, we describe progress in constructing and testing a state-of-the-art automated system for stereo compilation, new approaches to extracting depth and structural information from imaged data, a knowledge-based system for feature extraction, and tools for scene modeling and interaction with a machine terrain data base.

1 INTRODUCTION

The goal of this research program is to obtain solutions to fundamental problems in computer vision; particularly to such problems as stereo compilation, feature extraction, and general scene modeling that are relevant to the development of an automated capability for interpreting aerial imagery and the production of cartographic products.

To achieve this goal, we are engaged in investigations of such basic issues as image matching, partitioning, representation, and physical modeling. However, high-level, high-performance vision requires the use of both intelligence and stored knowledge (to provide an integrative framework), as well as an understanding of the physics and mathematics of the imaging process (to provide the basic information needed for a reasoned interpretation of the sensed data). Thus, a significant portion of our work is devoted to developing new approaches to the problem of "knowledge-based vision." Finally, vision research cannot proceed without a means for effective implementation, demonstration, and experimental verification of theoretical concepts; we have developed an environment in which some of the newest and most effective computing instruments can be employed for these purposes.

The research results described in this report are partitioned into three topic areas: (1) three-dimensional scene modeling and stereo reconstruction; (2) feature extraction; scene partitioning and semantic labeling; and (3) interactive scene modeling and knowledge-base construction.

2 THREE-DIMENSIONAL SCENE MODELING AND STEREO RECONSTRUCTION

Our goal in this research area is to develop automated methods for producing a 3-D scene model from several images recorded from different viewpoints. The standard approach to this problem is to use stereo compilation — a technique that involves finding pairs of corresponding scene points in two images (which depict the scene from different spatial locations) and using triangulation to determine scene depth. Various factors associated with viewing conditions and scene content can cause the matching process to fail; these factors include occlusion, projective or imaging distortion, featureless areas, and repeated or periodic scene structures. Some of these problems can only be solved by providing the machine with a global context for dealing with the missing or ambiguous information. Thus, an important component of this research effort, discussed in the section on interactive scene modeling, is to devise machinery by which a human operator can simply and effectively provide the needed information. In the remainder of this section we limit our discussion to direct approaches — more effective methods for image matching, interpolation for filling in "holes" caused by matching failure, and some exciting and radically new methods for 3-D modeling.

2.1 Baseline Stereo System

As a framework for integration and evaluation of our research in modeling 3-D scene geometry, as well as a vehicle for technology transfer, we have implemented a complete "state-of-the-art" stereo system. This system, described in Hannah [6] [7], is capable of producing a dense 3-D scene model from stereo pairs of intensity images. Included in these references are results of testing the system on a number of significant data sets. We believe that the current version of this fully automatic system is comparable to the best of the semiautomatic (human-assisted) systems now in operational use.

2.2 New Methods for Stereo Compilation

As previously indicated, the conventional approach to recovering scene geometry from a stereo pair of images is based on the matching of distinctive scene features, as well as on the satisfaction of constraints imposed by the viewing geometry (e.g., the epipolar constraint). Typically, three steps are required:

(1) determination of the relative orientation of the two images, (2) computation of a sparse depth map, and (3) derivation of a dense depth map for the given scene.

In the first step, points corresponding to unmistakable scene features are identified in each of the images. The relative orientation of the two images is then calculated from these points. This is, in part, an unconstrained matching task. Corresponding image features must be found. Without a priori knowledge, such a matching procedure knows neither the approximate location (in the second image) of a feature found in the first image, nor the appearance of that feature. However, it is often the case that appearance will vary little between images and that they were taken from similar positions relative to the scene.

Recovery of the relative orientation of the images reduces the computation of a sparse depth map from unconstrained two-dimensional matching to constrained one-dimensional matching. The quest for a scene feature identified in the first image is reduced to a one-dimensional search along an (epipolar) line in the second image. Identification of this feature in the second image makes it possible to calculate the feature's disparity and, hence, its relative scene depth.

Identification of corresponding points in the two images is typically based on correlation techniques. Area-based correlation processes may be applied directly to the raw image irradiances or to images that have been preprocessed in some manner. Edges (identified by the zero crossings of the Laplacian of their image irradiances) have also been used to obtain correspondences.

The outcome of this second step is a sparse map of the scene's relative depth at those points that were identified in both images of the stereo pair.

A sparse depth map does not define the scene topography. The third and final step in recovering the topography of the scene is "filling in" this sparse map to obtain a dense depth map of the scene. Typically, a surface interpolation or approximation method is used as a means of calculating the dense depth map from its sparse counterpart. A surface approximation model may be formulated to provide desirable image properties (such as the lack of additional zero crossings -- in the Laplacian of the image irradiances -- that are artifacts of the surface approximation model), but, often, the surface model is based on a priori requirements for the fitted surface, such as smoothness.

The problems encountered in the first two steps -- recovery of the relative orientation of the images and computation of the sparse depth map -- are dominated by the problems of image matching. False matches that arise from repetitive scene structures, such as windows of a building, or from image features that are not distinctive (at least on the basis of local evidence) occur more frequently in the unconstrained matching environment than in the constrained environment. In recovering the relative orientation of the images, we can use redundant information in an effort to reduce the influence of false matches; this is more difficult in the case when the sparse depth map is computed. Furthermore, we have little choice as to which features we may use for sparse depth mapping: if we choose not to use a feature, we cannot recover the relative depth at that scene point (without invoking semantic or contextual knowledge).

The selection of suitable features for determining image correspondence is difficult in itself. Correlation techniques embed assumptions that are often violated by the best image features. Area-based correlation techniques usually reflect the premise that image patches are of a scene structure that is positioned at one distinct depth, whereas edges that arise at an object's boundaries are surrounded by surfaces at different scene depths. Edge-based techniques are based on the assumption that an edge found in one image is not "moved" by the change in viewing position of the second image, whereas zero crossings found at boundaries of objects whose surface gradients are tangential to the line of sight contradict this assumption. These would seem minor problems, were it not for the accuracy required of the matching process. Often, the spatial resolution of disparity measurements must be better than the image's spatial resolution. Stereo matching sometimes requires features with properties that are incompatible with what is practical in realistic situations.

The third step, derivation of a dense depth map from a sparse one, is still far short of having an adequate solution. Most approaches employ "blind" interpolation, since no effective methods are currently in use for extracting depth from the irradiance data in the individual images of the stereo pair.

In summary, we see that the most demanding steps in the stereo process are the final two: computation of a sparse depth map, and derivation of its dense counterpart. In Smith [13], we describe a new approach to stereo compilation that involves combining these steps to recover a dense relative-depth map of the scene directly from the image data. We use image irradiance profiles as input to an integration routine that returns the corresponding dense relative-depth profile. This procedure neither matches image points (at least not in the conventional sense), nor does it "fill in" data to obtain the dense depth map. It avoids the need to make the restrictive assumptions usually required for stereo image matching, and it directly uses the image irradiance data in recovering the dense depth map.

2.3 New Methods for 3-D Modeling Using Methods Which Do Not Depend On Stereo Correspondence

We have noted the fact that it will not always be possible to find corresponding scene points in the two images of a conventional stereo pair, and yet, to recover a dense scene model, we need to determine the depth at every scene point. Since interpolation will not always provide an acceptable answer when matching fails, we are investigating a number of new techniques for recovering scene depth that do not require establishing stereo correspondence.

A significant body of work exists in the area of extracting depth from the shading and texture visible in a single image. However, these different techniques make a variety of distinct assumptions about the nature of the scene, the illumination, and the imaging geometry. In Strat and Fischler [14], we show that the distinct assumptions employed by each of these different schemes must be equivalent to providing a second (virtual) image of the original scene, and that all of these different approaches can be translated into a conventional stereo formalism. In particular, we show that it is frequently possible to structure the problem as that of recovering depth from a stereo pair consisting of a conventional perspective image (i.e., the original image) and an orthographic image (the virtual image). We also provide a new algorithm needed to accomplish this type of stereo-reconstruction task.

In Pentland [10] we show how focal gradients (image "blur"), resulting from the limited depth of field inherent in most optical systems, can be used to recover scene depth. The advantages of this technique are that it is fast, computationally simple, makes no special assumptions about the scene, and avoids the stereo-matching problem. Mathematical analysis and experiments indicate that the accuracy achievable by this technique is comparable to what can be expected from the use of stereo disparity or motion parallax in determining scene depth.

For most purposes concerned with the analysis of imaged data, determination of an array of depths (e.g., as obtained by conventional stereo methods) is only the first step in the construction of a scene description. The conventional approach next compiles largely continuous surfaces from the discrete depth information and then attempts to partition these surfaces into coherent 3-D objects. Aside from some still unsolved theoretical problems, this process is computationally expensive and time consuming. In Bolles and Baker [2], we describe a new method for using camera motion through a scene to obtain a 3-D model in which higher level scene attributes are directly accessible. This technique is based on considering a dense sequence of images as forming a solid block of data. Slices through this solid at appropriately chosen angles intermix time and spatial data in such a way as to simplify the partitioning problem: these slices have more explicit structure than the conventional images from which they were obtained. We believe that this work is a very important development; it offers a completely new and direct method for accessing information about scene objects without requiring a completely bottom-up analysis process.

3 FEATURE EXTRACTION: SCENE PARTITIONING AND SEMANTIC LABELING

Creating a scene description from a photographic image requires the ability to perform two basic operations: (a) partitioning the image into independent or coherent pieces, and (b) assigning names or semantic labels to these pieces.

The partitioning operation, necessary to reduce the computational complexity of the subsequent scene-analysis steps, has proven to be extremely difficult to accomplish: the performance of automated systems is still far inferior to that of humans. In part, this disparity in performance occurs because humans appear to employ contextual knowledge and past experience in such tasks, while most available computational techniques employ only the local intensity patterns visible in the image, i.e., they perform "syntactic partitioning." For practical as well as theoretical reasons, we have been pursuing an investigation (1) to determine the competence limits of a purely syntactic approach to partitioning and, simultaneously, (2) to construct an operational system that approaches these limits. This investigation is nearing completion and has resulted in a very high performance system that will be described in a paper by Laws now in preparation.

In Barnard [1], we describe one of a number of on-going investigations that attempt to provide a theoretical basis for the partitioning process. In this paper, Barnard explores the idea that partitioning decisions result in alternative descriptions of a scene, and that the preferred partitioning is the one that provides the "simplest" description. In a paper by Fischler and Bolles [3], partitioning is viewed as an explanation of how the image is related to the scene from which it was derived; it is shown that completeness and stability of explanation, as well as simplicity, are useful partitioning criteria since these attributes are necessary for an explanation to be believable.

In Fua and Hanson [5], we describe an approach to the problem of converting a syntactically partitioned image (e.g., one provided by Laws' segmentation system [8]) into a semantic description. This work has resulted in a system that can extract cultural objects from aerial imagery; it employs geometric reasoning to identify semantically significant arrangements of straight line segments in the borders of the supplied partition. Emphasis is placed on using generic models characterizing significant kinds of geometric relationships and shapes, thereby avoiding the well-known drawbacks inherent in the use of specific object templates. An important feature of the system (still under development) is the generation of an explanation for any detected discrepancy between the hypothesized object models and the initial partition. In principle, this technique should permit intelligent compensation for anomalies due to imaging or environmental effects that would be recognized by a well-briefed human analyst; for example, the system should be able to identify two contrasting regions of a peaked roof as belonging to a single house based on illumination effects consistent with the known sun position. The ability of this system to explain its decisions in terms of deviations of sensed data from stored models appears to offer an effective mechanism for understanding the operation of the system and, simultaneously, a basis for improving its performance.

4 INTERACTIVE SCENE MODELING AND KNOWLEDGE-BASE CONSTRUCTION

Our intent in this effort is to develop a system framework for allowing higher-level knowledge to guide the detailed interpretation of imaged data by autonomous scene-analysis techniques. Such an approach allows symbolic knowledge, provided by higher-level knowledge sources, to control automatically the selection of appropriate algorithms, adjust their parameters, and apply them in the relevant portions of the image. More significantly, we are attempting to provide an efficient means for supplying and using qualitative knowledge about the semantic and physical structure of a scene so that the machine-produced interpretation, constrained by this knowledge, will be consistent with what is generally true of the overall scene structure, rather than just a good fit to locally applied models.

An important component of our approach is to design a means for a human operator simply and effectively to provide the machine with a qualitative scene description in the form of a semantically labeled 3-D "sketch." This capability for effective communication between a human and a machine about the three-dimensional world requires both appropriate graphics tools and an ability on the part of the machine for both spatial reasoning and some semantic "understanding." The importance of this work derives from the fact that a major difficulty in automating the image-interpretation process is the inability of current

computer systems to deduce, from the visible image content, the general context of the scene (e.g., urban or rural; season of the year; what happened immediately before, and what will happen immediately after, the image was viewed by the sensor) -- the knowledge-base and reasoning required for such an ability is well beyond what the state of our art can hope to accomplish over (at least) the next 5 years. Thus, our work is intended to provide a means by which a human can supply, to a task-oriented program, the high-level overview the program needs for its analysis of a given scene, but cannot acquire by itself.

4.1 The Representation and Modeling of Natural Forms

Our research in this area addresses three related problems: (1) representing natural shapes such as mountains, vegetation, and clouds; (2) computing such descriptions from image data; and (3) interactively providing the machine with a description of natural forms as a way of building an internal knowledge data base. The first step towards solving these problems is to obtain a model of natural surface shapes.

A model of natural surfaces is extremely important because we face problems that seem impossible to address with standard descriptive computer-vision techniques. How, for instance, should we describe the shape of leaves on a tree? Or grass? Or clouds? When we attempt to describe such common, natural shapes using standard representations, the result is an unrealistically complicated model. Furthermore, how can we extract 3-D information from the image of a textured surface when we have no effective models that describe natural surfaces and how they evidence themselves in the image? The lack of such a 3-D model has restricted image texture descriptions to being ad hoc statistical measures of the image intensity surface.

Fractal functions, a novel class of naturally arising functions, are a good choice for modeling natural surfaces because many basic physical processes (e.g., erosion and aggregation) produce a fractal surface shape, and because fractals are widely used as a graphics tool for generating natural-looking shapes. Additionally, in a survey of natural imagery, we found that a fractal model of imaged 3-D surfaces furnishes an accurate description of both textured and shaded image regions, thus providing validation of this physics-derived model for both image texture and shading.

Progress relevant to computing 3-D information from imaged data by use of a fractal model is described in Pentland [9]. A test has been derived to determine whether or not the fractal model is valid for a particular set of image data, an empirical method for computing surface roughness from image data has been developed, the computation of a 3-D fractal-based representation from actual image data has been demonstrated, and substantial progress has been made in the areas of shape-from-texture and texture segmentation. Characterization of image texture by means of a fractal surface model has also shed considerable light on the physical basis for several of the texture-partitioning techniques currently in use and has made it possible to describe image texture in a manner that is stable over transformations of scale and linear transforms of intensity.

In [11], Pentland describes an interactive system for modeling natural forms. This system employs superquadrics, as well as fractal functions, in allowing the user simply and effectively to create and display almost any iconic object (e.g., the human form, surfaces with analytic descriptions, natural terrain, etc.).

This research is expected to contribute to the development of (1) a computational theory of vision applicable to natural surface shapes, (2) compact representations of shape useful for describing natural surfaces, and (3) real-time modeling, generation, and display of natural scenes. We also anticipate adding significantly to our understanding of the way humans perceive natural scenes.

4.2 Interactive Modeling and Analysis via Machine Synthesized Imagery

Terrain-Calc, described in Quam [12], is a system for synthesizing realistic sequences of perspective stereo views of real-world terrain (described within the machine by a database of geometric and photometric models). This system, implemented on a Symbolics 3600 Lisp Machine, has a sophisticated graphical interface, which allows the user to specify an arbitrary flight path over a modeled piece of terrain. A sequence of views (single images or stereo pairs, as desired), spaced at equal distances along the flight path, is generated at about one frame per minute, and up to 60 frames can be displayed at a rate of sixteen frames per second. This system is revolutionary in its flexibility, computational efficiency, and the quality of the renderings it produces, given that it does not employ any special-purpose hardware.

4.3 Architectures for Interactive and Real-Time Machine-Vision Systems

The computational demands imposed by interactive and real-time, machine-vision applications frequently exceed the capacity of conventional computer architectures. For this reason, attempts have been made to reduce computation time by decomposing serial algorithms into segments that can be simultaneously executed on parallel hardware architectures. Because many classes of algorithms do not readily decompose, one seeks some other basis for parallelism. In Fischler and Firschein [4] we show (1) that "guessing" the answer to a problem and then checking its validity is a useful approach, and (2) that a number of vision algorithms are based on this concept. A parallel architecture capable of executing such algorithms is proposed.

5 ACKNOWLEDGEMENT

The following researchers have contributed to the work described in this report: H. Baker, S. Barnard, J. C. Bolles, O. Firschein, M. A. Fischler, P. Gai, M. J. Hannah, A. J. Hanson, D. L. Kashtan, K. Laws, A. P. Pentland, L. H. Quam, G. B. Smith, T. Strat, and H. C. Wolf.

References

[1] Barnard, S.T., "An Inductive Approach to Figural Perception," AIC Technical Note 325, SRI International, Menlo Park, California (September 1984).

[2] Bolles, R.C. and H.H. Baker, "Epipolar-Plane Image Analysis: A Technique for Analyzing Motion Sequences," to appear in the *Third International Symposium on Robotics Research*, Paris, France (October 1985); also these proceedings.

[3] Fischler, M.A. and R.C. Bolles "Perceptual Organization and Curve Partitioning," *Proceedings of the 1983 Image Understanding Workshop* (June 1983); also IEEE CVPR-83.

[4] Fischler, M.A. and O. Firschein, "Parallel Guessing: A Strategy for High-Speed Computation," AIC Technical Note 338, SRI International, Menlo Park, California (September 1984).

[5] Fua, P.V., and A.J. Hanson, "Locating Cultural Regions in Aerial Imagery Using Geometric Cues," these proceedings.

[6] Hannah, M.J., "Evaluation of STEREOSYS vs. Other Stereo Systems" and "The Stereo Challenge Data Base," AIC Technical Notes 365 and 366, respectively, SRI International, Menlo Park, California (October 1985).

[7] Hannah, M.J., "SPI's Baseline Stereo System," these proceedings.

[8] Laws, K.I., "Goal-Directed Textured-Image Segmentation," Technical Note 334, SRI International (September 1984).

[9] Pentland, A.P., "Fractal Based Description of Natural Scenes," *Proceedings of the 1983 Image Understanding Workshop* (June 1983); also IEEE CVPR-83.

[10] Pentland, A.P., "A New Sense for Depth of Field," *Proceedings of International Joint Conference on Artificial Intelligence*, Los Angeles, California (August 1985).

[11] Pentland, A.P., "Perceptual Organization and the Representation of Natural Form," AIC Technical Note 357, SRI International, Menlo Park, California (July 1985).

[12] Quam, L.H., "The Terrain-Calc System," these proceedings.

[13] Smith, C.B., "Stereo Reconstruction of Scene Depth," *Proceedings of Computer Vision and Pattern Recognition '85*, San Francisco, California (June 19-23, 1985).

[14] Strat, T.M. and M.A. Fischler, "One-Eyed Stereo: A General Approach to Modeling 3-D Scene Geometry," *Proceedings of International Joint Conference on Artificial Intelligence*, Los Angeles, California (August 1985); also these proceedings.

GC970145-08

Image Understanding Research at Columbia

John R. Kender[1]

Department of Computer Science
Columbia University, New York, NY 10027

0 Abstract

The Computer Vision Laboratory at Columbia University continues to focus on problems of middle-level vision. Our recent emphasis has been on the complexity of image understanding tasks, and on the implementation of middle-level algorithms on parallel processors. We have derived several new algorithms for shape-from methods, including a provably convergent algorithm for shape-from-shading and an efficient algorithm for smoothing the optic flow field. We have demonstrated that the depth interpolation problem is critically dependent on definitions of "smoothness" and have derived at least four major classes of algorithms for its solution in the context of computer vision. Two of these we have critically compared. We have begun to explore algorithms on SIMD machines that are computationally and communicationally efficient for the derivation of sparse depth from stereo, and for the derivation of full depth from sparse depth.

In work on high level-vision, we have demonstrated some theoretic deficiencies in the generalized cylinder approach to object modelling. Additional work on middle- and high-level vision progresses-- on texture and texture gradients, on object modelling and qualitative shape description, and on natural language correlates to qualitative aspects of spatial relations.

1 Introduction

The Computer Vision Laboratory at Columbia continues its steady growth. A second professor of computer science (Peter Allen) and a research associate (Hussein Ibrahim) now share in directing the work. Faculty, staff, and students have reached 13 people. Our VAX has been augmented with three Sun workstations, and our Puma 560 robotic arm and our Matrix color film recorder are both up and running. We have graduated another Ph.D. student (David Lee), and can reasonably expect at least one other in the coming year.

Our research investigations fall into the following categories.

- The analysis of the complexity of middle-level vision algorithms, including depth approximation, shading, and optic flow, both abstractly via mathematics and experimentally via psychology (David Lee and Terry Boult).

- The parallelization on mesh- and tree-connected SIMD machines of middle- and high-level vision algorithms, including stereo, depth approximation, and model matching using extended Gaussian images (Hussein Ibrahim and Dong Choi).

- The investigation of shape-from-texture algorithms, including the analysis of textures under illumination and viewpoint changes, and the exploitation and coordination of multiple texture knowledge sources (Mark Moerdler, Paul Douglas, and George Wolberg).

- The analysis and implementation of existing techniques for quantitative and qualitative three-dimensional shape analysis, with special attention to aspect graphs and generalized cylinders (David Freudenstein, Ken Roberts, and Ari Gross).

- The generation of natural language text from a representation of spatial relationships, concentrating on the semantics of words denoting relative and absolute scaled quantities (Michal Blumenstyk).

- Usual system support activities, including the investigation of efficient half-toning techniques for various hard copy devices (Earl Smith, laboratory manager, and Dina Berkowitz, staff).

2 Complexity of Image Algorithms

Using the methods of both numerical analysis and information based complexity, we have attempted to quantify the difficulty of middle-level vision tasks and to provide optimal algorithms for their solution. We have explored several such problems, including depth interpolation, shape from shading, and the smoothing of the optic flow field. Further, we have investigated several issues related to optimality, including: the precise definition of a "smooth" solution to depth, shading, and flow; the human psychophysical perceptibility of "smoothness"; the computational efficiency of our algorithms relative to existing methods, and the information gathering circumstances under which any of the several "smooth" solutions is the most appropriate.

2.1 Depth, Shading, and Optic Flow

We have demonstrated that the interpolation or approximation of full depth values from the sparse depth values derived from stereo or other means is optimally solved (in the worst case) by splines [9, 14]. Depending on the imaging situation, this implies that the full depth map can be obtained in time only linear in the data. Further, we showed that under many imaging conditions, heuristic adaptation cannot affect the speed or precision of the result; the computation can be decision-free, with no loss of accuracy. We also noted that many of the algorithmic implementations of this abstract result would be especially blessed with seperabilities and symmetries, and so would be well suited for hosting on a SIMD machine: a conclusion we are vigorously pursuing in practice (see Section 3).

[1]This work was supported in part by the Defense Advanced Research Projects Agency under contract # 00039-84-C-0165, and in part by an NSF Presidential Young Investigator Award

This work highlighted the disturbing lack of justification for the invocation of any particular class of functions as being a useful model for real world objects. The mathematical approach we used was general enough so that the conclusion regarding splines was valid even over a wide range of object models, but the definition of just what is being sought remains a free parameter in the approach. Evidence for supporting one's real world assumptions of "smoothness" must be gathered extrinsically, a topic which we are investigating also.

In work on shape from shading, we have reformulated the mathematical basis for the relaxation methods of Ikeuchi and Horn. This has lead to a new algorithm whose convergence, under certain general circumstances, is provable [14, 15]. The solution again is cast in terms of smoothing splines, under a straightforward definition of "smooth" surface.

The solution is unique, iteratively obtainable, and guaranteed to converge if the reflectance function sufficiently models the surface properties and if the image intensity measurements are sufficiently precise. These two "sufficiently"s are quantifiable; they appear in the analysis as limits on the penalty parameter lambda used to weight the relative significance of reconstructed image accuracy versus reconstructed image smoothness. We obtained limits for this parameter in the abstract; in general, the more variation there is in the reflectance map, the less the accuracy that can be demanded of the reconstruction and still have convergence guaranteed. In the case of Lambertian reflectance, we give a numeric range.

In addition, the algorithm has very attractive complexity. Because it is dominated by the multiplication of a vector by a sparse matrix with special properties, the fast Fourier transform can be used instead of the conventional method. To obtain an image reconstruction with accuracy to $O(1/N)$ (within the range allowable), the algorithm runs in $O(N^2 \log^2 N)$ steps, where NxN is the number of pixels. (We have yet to test the algorithm on real images.)

In work on smoothing the optic flow field, we applied similar techniques to the methods of Horn and Schunck, and of Cornelius and Kanade [15]. Again using smoothing splines, we can show that within the unit square, the matrix central to the method is symmetric positive definite. We were able to bound its minimum and maximum eigenvalues, which suggests that the Chebyshev iterative method of solution would be most appropriate and quick to converge. The proposed algorithm we have shown must a unique solution, although we are as yet unable to guarantee convergence. Like our work on shape from shading, the algorithm employs the FFT in lieu of the standard method for multiplying a vector by the matrix, so it too has low computational complexity.

Future work in this area, including implementation, is detailed in [13].

2.2 Smoothness and Efficiency

We have investigated the ubiquitous problem of image property smoothness. In much of the work on middle-level vision, it is necessary to define the desired "smoothness" for the solution manifold: whether for an object surface or for a retinal vector field. However, it seems that current definitions usually have little a priori basis; often it appears that what is assumed is what is *convenient* to assume, rather than what *ought* to be assumed.

One justification for a definition of smoothness is that it corresponds to the apparent limits of human beings to detect changes in higher order derivatives. We investigated what is known about these limits [1], but there is surprisingly little in the literature, and most of it is on smooth *motion* perception. Nevertheless, we showed that psychologists who posit formal models of the human vision system are forced to assume--usually implicitly--specific limits. But they do not agree, and they invoke a wide range of differentiabilities for "smooth" objects in the world, for the "smooth" images they cause, and for the "smooth" motions they trace. We demonstrated that this disagreement is reflected in our own field; again, a wide range of differentiability is assumed in work on depth reconstruction, shape from shading, and optic flow.

In work on surface fitting, we have stressed that "smoothness" is not simply a matter of semantics; the definition usually has a profound effect on the quality, accuracy, and complexity of the algorithms necessary to achieve it. We have shown that it is meaningful to say "smooth" in at least four fundamentally different ways [2]. Each of these ways invokes different assumptions: for example, band-limiting the surface, having smoothness be non-isotropic, incorporating a priori knowledge, etc. We have discussed the often overlooked difference between requiring surfaces to interpolate given depth data versus merely to approximate it; more is known about the complexity of the former. We have illustrated that any of these definitions can realized in one of four basic ways: by classical minimization, by multigrid methods, by two types of reproducing kernel splines. We detailed the advantages and disadvantages of each realization, and gave guidelines on how to choose among them under different imaging situations.

In related work, we have constructed a reference catalog for the reproducing kernels needed in the spline approach [3]. In it we gave implementation details and some examples for the algorithms that arise under the four definitions of "smooth" already referred to, as well as under four additional definitions.

Lastly, we have compared Grimson's gradient-projection approach for depth interpolation to two variants of our method based on splines [4]. We derived order estimates for all three algorithms' serial time complexity, serial space complexity, and parallel time complexity assuming a SIMD machine. We pointed out that the spline methods always calculate a unique solution surface, but that the gradient-projection may not converge. Further, we indicated other advantages of spline methods. They produce a function that can give, along with the dense depth values, valuable surface properties such as the gradient or local "smoothness". We found that spline methods are locally calculable, isotropic, and can serve as compact surface descriptors. We also showed that, unlike the gradient-projection method, they are easily extensible to many definitions of "smooth".

3 Parallelization of Image Algorithms

We have continued to analyze and encode vision algorithms to be hosted by Columbia's fine-grained, tree- and mesh-connected, SIMD parallel processor, Non-Von. A prototype of 63 nodes has been running since early 1985, when we demonstrated the segmentation of real image data using quad trees. Most of our work continues in the abstract, with algorithms verified by a functional simulator. We have examined algorithms at low- and middle-level, and have begun work on a high-level image analysis task.

3.1 Low and Middle Level Tasks

We have demonstrated that Non-Von is a cost-effective processor for low level vision tasks [7]. We have designed and tested algorithms for binary image tree and quad tree creation and manipulation, for image correlation, histograming, and connected component labeling, and for geometric property computations such as moments, compactness, and Euler number. The encoding of the algorithms incorporated novel approaches to control flow that reduced the effects of the communication bottleneck usually associated with tree architectures. We have recently begun to design and test algorithms for stereo.

At the middle level, we have described, simulated, and analyzed the performance of algorithms for a representative Hough transform, and for an algorithm that is used in the interpretation of moving light displays [8]. We concluded that even in middle level vision it is possible to exploit the

available massive parallelism of a SIMD machine. We showed that by carefully and inexpensively duplicating data and/or control information, and by delaying or avoiding the reporting of intermediate results, it is possible to avoid many of the communication bottlenecks otherwise common at this level of image understanding.

3.2 Middle and High Level Tasks

Emboldened by our initial success, we have been to investigate the design and execution on Non-Von of two image understanding tasks at higher levels still. The first is the depth interpolation problem, whose solution via the adaptive Chebyshev acceleration method appears to be a natural fit to the tree- and mesh-connections of the machine [6]. We have simulated several small interpolations, and compared our speed and accuracy with existing iterative methods based on Gauss-Seidel. We have begun to investigate the performance of two other similar methods: the pure Chebyshev, and the conjugate gradient method.

The second task is still preliminary, but it is probably rightly called a high level vision problem. We are designing ways in which Non-Von can parallelize the recognition of objects from their extended Gaussian images. We hope to demonstrate that SIMD architectures can be used in model matching, and might even profit from the use of heuristic search in their control structures.

4 Analysis of Texture

We have begun an exploration of the properties of fractal textures under imaging assumptions which are more realistic than those currently in use. In particular, we are examining the effects that oblique illumination or view angles have on the fractal dimension of the imaged texture. We wish to understand these effects enough to invert them, and to derive from them constraints on surface orientation and distance.

In applied work on texture, we continue to exploit and coordinate multiple shape-from-texture knowledge sources [12]. One source is based on the virtual lines (spacings) that occur between line-like elements [10]. We have shown that it is quite robust in the presence of noise and texel perturbation; it is even capable of handling primitive forms of object transparency and virtual surfaces. We continue to expand the scope of the system, and hope to program and integrate a module that exploits the constraints implicit in gravitationally-based environmental labels such as "horizontal" and "vertical" [11].

5 Analysis of Three-Dimensional Shape

As part of our initial incursion into the study of three-dimensional representations for shape, we surveyed systems that explicitly capture and/or internally maintain depth measurements [5].

Delving further, we have demonstrated the difficulty of using generalized cylinders as a canonical representation for object models. Building on work of Shafer, we have shown that even when only straight homogeneous generalized cylinders are considered, theorems about the uniqueness of descriptions of a given shape are difficult to prove. Regular polyhedra and simple egg-shaped objects are counterexamples to even minor attempted extensions to an existing theorem [16].

6 Generating Text about Spatial Relations

In a new investigation, we have begun to analyze the representations and state information that are required for the generation of natural language descriptions of spatial relations. In particular, we have noted that many statements regarding measurable physical quantities such as length, mass, or time are given qualitatively with respect to an assumed focal point or reference. Thus, trees are "nearby", or houses are "too big". We hope to obtain sufficient insight into the presumed topological representations underlying such utterances that it can contribute to our efforts on three-dimensional shape.

References

1. Boult, T. Smoothness Assumptions in Human and Machine Vision, and Their Implications for Optimal Surface Interpolation. Department of Computer Science, Columbia University, 1985.

2. Boult, T., and Kender, J. On Surface Reconstruction Using Sparse Depth Data. Proceedings of the ARPA Image Understanding Workshop, Dec., 1985. (These proceedings.)

3. Boult, T. Reproducing Kernels for Visual Surface Interpolation. Department of Computer Science, Columbia University, 1985.

4. Boult, T. Visual Surface Interpolation: A Comparison of Two Methods. Proceedings of the ARPA Image Understanding Workshop, Dec., 1985. (These proceedings.)

5. Boult, T. A Survey of Some Three-Dimensional Vision Systems. SIGART Newsletter, April, 1985, pp. 28-37.

6. Choi, D., and Kender, J. Solving the Depth Interpolation Problem with the Adaptive Chebyshev Acceleration Method on a Parallel Computer. Proceedings of the ARPA Image Understanding Workshop, Dec., 1985. (These proceedings.)

7. Ibrahim, H.A.H., Kender, J.R., and Shaw, D.E. The Analysis and Performance of Two Middle-Level Vision Tasks on a Fine-Grained SIMD Tree Machine. Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, June, 1985, pp. 248-257. (An expanded version has been submitted to *Computer Vision, Graphics, and Image Processing.*)

8. Ibrahim, H.A.H., Kender, J.R., and Shaw, D.E. "Low-Level Image Understanding Tasks on Fine-Grained Tree-Structured SIMD Machines." *Journal of Parallel and Distributed Computing* (Submitted for publication 1985).

9. Kender, J.R., Lee, D., and Boult, T. Information-Based Complexity Applied to Optimal Recovery of the 2-1/2 D Sketch. Proceedings of the Third Workshop on Computer Vision : Representation and Control, IEEE Computer Society, Oct., 1985, pp. 157-167.

10. Kender, J.R., and Moerdler, M. Surface Orientation and Segmentation from Perspective Views of Parallel-Line Textures. Department of Computer Science, Columbia University, Jan., 1985.

11. Kender, J.R. "Environmental Labelings in Low-Level Image Understanding." *Artificial Intelligence* (In revision 1985).

12. Kender, J.R., *Artificial Intelligence Research Notes*, Volume : *Shape from Texture*. Pitman Publishing, Ltd., London, To appear 1986.

13. Lee, D. "Optimal Algorithms for Image Understanding : Current Status and Future Plans." *Journal of Complexity* (To appear 1986).

14. Lee, D. *Contributions to Information-based Complexity, Image Understanding, and Logic Circuit Design*. Ph.D. Th., Department of Computer Science, Columbia University, Sept. 1985.

15. Lee, D. A Provably Convergent Algorithm for Shape from Shading. Proceedings of the ARPA Image Understanding Workshop, Dec., 1985. (These proceedings.)

16. Roberts, K. Equivalent Descriptions of Generalized Cylinders. Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, June, 1985. (Also these proceedings.)

GC9761458

SUMMARY OF PROGRESS IN IMAGE UNDERSTANDING AT THE UNIVERSITY OF MASSACHUSETTS

Edward M. Riseman and Allen R. Hanson

Computer and Information Science Department
University of Massachusetts
Amherst, Massachusetts 01003

ABSTRACT

This research summary documents several areas of research at the University of Massachusetts that are entirely or partially supported under the DARPA image understanding program. The work is divided into several areas: motion analysis, low-level and intermediate-level processing, and knowledge-based processing strategies. Some of this work is documented in several papers in these proceedings.

I. MOTION ANALYSIS

Our research in motion analysis continues to broaden with research in several theoretical and experimental areas.

I.1. EFFECTIVENESS IN RECOVERING TRANSLATIONAL MOTION PARAMETERS

We have continued the analysis of algorithms for constrained sensor motion [LAW84]. In particular we are evaluating the robustness, accuracy, and efficiency of the algorithm for recovering translational motion parameters [PAV85]. Here the global search for the focus-of-expansion (FOE) requires the computation of the sum of errors (e.g., via correlation) associated with the displacement of a set of feature points in two or more frames. A sparse sampling of the possible location of FOEs provides a global error function whose minimum localises the direction of motion.

The accuracy and robustness is a function of the number of points that are tracked and contribute to the error function, which of course must be traded off against the amount of computation that can be tolerated for real-time motion analysis. Thus far, our experiments on simulated environments imply that there is a wide range of situations for which the motion parameters can be approximately recovered at relatively modest computational expense. Specifically, when the angle between the image plane and the direction of translational motion is less than 60 degrees, then between 4 and 16 points which are widely spaced in the image are sufficient to recover the approximate motion of the sensor. A smaller number of points (4–8 points) is necessary when the camera is oriented approximately in the direction of motion, and a larger number of points (8–16) when the camera orientation is at a modest angle (15 degrees to 45 degrees) with respect to translation. When the angle between camera orientation and translation is large (60 degrees to 90 degrees) there appears to be a flat error surface around the correct motion, leaving a wide range of ambiguity no matter how many feature points are employed. This result not surprising in that it states that when a camera is pointing out the driver's side window, accurately determining the motion of a vehicle moving down the road is not possible.

This work was supported by DARPA under Contract N00014-82-K-0464.

I.2. INHERENT AMBIGUITY IN MOTION ANALYSIS OF NOISY FLOW FIELDS

In the cases where the sensor motion is unconstrained and/or there are independently moving objects in the environment, our algorithms for direct recovery of motion parameters and environmental structure are not applicable. Therefore, we turn to the usual method of motion analysis which is decomposed into two phases: computation of an optical flow field and interpretation of this field. In the present discussion, the term "optical flow field" refers to a "velocity field", composed of vectors describing the instantaneous velocity of image elements. The computation of reliable flow fields is the subject of work presented in Section I.3. The second phase, which is the general interpretation of flow fields, was the subject of previous work by Adiv [ADI85a,b].

The work discussed in this section mathematically examines the robustness of algorithms for interpreting general motion from flow fields. The analysis focusses on ambiguities that are inherent in the sense that they are true of all algorithms, and can only be resolved if constraining assumptions or other sources of visual information are employed.

Two problems which may arise due to the presence of noise in the flow field have been examined. Since noise in flow fields must be expected almost always to be present, we believe this analysis is relevant to all real situations of motion interpretation.

The first ambiguity is in recovering the motion parameters from a noisy flow field generated by a rigid motion. Motion parameters of the sensor or a rigidly moving object

may be extremely difficult to estimate because there may exist a large set of significantly incorrect solutions which induce flow fields similar to the correct one. We found that if the field of view corresponding to the region containing the interpreted flow field is small, and the depth variation and translation magnitude are small relative to the distance of the object from the camera, then the determination of the 3-D motion and structure can be expected to be very sensitive to noise and, in the presence of a realistic level of noise, practically impossible. We experimentally found that there is also a relation between the location of the FOE and the degree of ambiguity.

The second ambiguity is in the decomposition of the flow field into sets of vectors corresponding to indepedently moving objects. The rigidity assumption [ULL79] has been found to be inappropriate for noisy flow fields; that is, the consistency of a set of flow vectors with the same motion parameters, up to the estimated noise level, does not reasonably guarantee that they are really induced by one rigid motion. Two independently moving objects may induce optical flows which are compatible with the same motion parameters and hence, there is no way to refute the hypothesis that these flows are generated by one rigid object. As an alternative to the usual rigidity assumption, it is assumed in [ADI85a,b] that a connected set of flow vectors, which is consistent with a rigid motion of a *planar* surface, is induced by a rigid motion. This assumption is weaker than the standard assumption in the sense that it can only be applied in more restricted situations and, therefore, it is more likely to be correct.

The results of the ambiguity analysis can be used when the effectiveness of motion algorithms is evaluated for real-world tasks. They can help to decide which algorithm to choose, and in what situations this algorithm can be expected to be effective. Recovering motion and structure of independently moving objects may be particularly difficult, as was demonstrated by the flat error surfaces obtained for such objects in the second and fifth experiments in [ADI85b]. In general, ambiguity in recovering 3-D motion and structure of independently moving objects can be expected, since the effective field of view and the ratio of the depth variation to the distance between the object and the camera are usually small. Even in ambiguous situations, constraints and parameters might be extracted. Integration of such partial information over a time sequence of flow fields may, eventually, resolve the ambiguity and result in a unique interpretation.

I.3. RELIABLE COMPUTATION OF OPTIC FLOW: A SMOOTHNESS CONSTRAINT AND A CONFIDENCE MEASURE

Although our hierarchical correlation algorithm [GLA83] for the computation of dense displacement fields has proved to be an efficient and reliable technique, there are still a number of situations where the algorithm makes mistakes. These situations arise in areas of images without significant intensity variations and at occlusion or motion boundaries. Our previous work [ANA84] attempted to identify such situations through the use of a confidence measure which indicated the reliability of a match vector. Our current work attempts to improve matches with low confidence based on neighbouring matches with higher confidences, by means of a relaxation process.

The confidence measure that was described in [ANA84] is a scalar value between 0 and 1 that indicated the reliability of the displacement vector at a pixel in the image. One such value was provided for each pixel. This measure was derived by studying the properties of the error-surface obtained during the process of computing the displacement at a pixel. However, the image displacement vector is a two-dimensional quantity. Hence, it is appropriate to have a two-dimensional confidence measure associated with the displacement vector.

In our previous work [ANA84], we observed that the error-surface allowed us to distinguish between situations where we had completely reliable information information regarding the displacement vector (i.e., at high curvature points along image contours), where we had partial information (i.e., at edge locations where only the displacement perpendicular to the edge can be reliably measured), and situations where we had no reliable information (at homogeneous intensity areas of the image). The new confidence measure is a vector quantity which uses these distinctions.

Our current work consists of two steps. The first is the computation of these vector valued confidence measures and the second is the smoothing process which corrects unreliable displacement vectors based on their reliable neighbours.

1. The new confidence measure is best described as a two-dimensional vector. It is convenient to describe the vector in terms of two orthogonal basis vectors e_{max} and e_{min}, which vary from pixel to pixel in an image. The displacement vector D can be decomposed in terms of its components along these basis vectors and confidence measures c_{max} and c_{min} are associated with these components. The basis vectors and the confidence measures can be easily understood by their behaviour at a high curvature point, an edge point and a point in a homogeneous area of the image.

At a high-curvature point both c_{max} and c_{min} will be high, indicating that all the components of displacement vector is highly reliable. In this case the exact directions of e_{max} and e_{min} are not crucial, and will depend on the precise shape of the contour. At an edge point c_{max} will be high and c_{min} low, and e_{max} and e_{min} will respectively be perpendicular and parallel to the edge. At a homogeneous area both the confidences will be low, and the directions of the basis vectors will depend on the details of the image intensity variations at that point.

Finally, the new confidence measures are also based on the shape of the correlation error surface. The details of their computation are described in [ANA85]. It is worthwhile to note that these are no longer bound to be between 0 and 1. The formulation of the smoothness constraint described below requires that these values be allowed vary between 0 and ∞.

2. The process of improving unreliable match estimates based on its neighbours is formulated as a smoothness constraint on the displacement vector field. The smoothness constraint consists of two errors E_{smooth} and E_{approx}, whose sum is minimised.

E_{smooth} measures the spatial variation of the displacement field - i.e., the smoother the variation, the smaller is the error. One example of such a constraint can be found in the work of Horn and Schunck [HOR81]. E_{approx} measures the deviation of the smooth displacement field from the initial field provided by the matching process.

$$E_{approx} = \sum c_{max}((U - D) \cdot e_{max})^2 + c_{min}((U - D) \cdot e_{min})^2$$

where U is the smoothed displacement vector and D is the initial veector at a pixel provided by the matching process. The definition of this error makes it clear that the low confidence estimates are allowed to vary more than the high confidence estimates. Hence, the smoothing process modifies the initial displacement values at locations low confidence measures more than those at the locations of high confidence measures.

The smoothness constraint translates into a minimisation problem. We solve this problem using the finite-element method, because this method permits the inclusion of known discontinuities in the displacement field. The application of this method leads to a local relaxation algorithm, which iteratively updates the displacement vector field.

Our future work will consist of developing techniques for locating the displacement discontinuities, gaining a greater understanding of the confidence measures (in particular how to normalise them) and possible improvements to the smoothness error.

I.4. REFINEMENT AND PREDICTION OF IMAGE DYNAMICS AND ENVIRONMENTAL DEPTH MAPS OVER MULTIPLE FRAMES

To a large extent research in the interpretation of motion has focussed on the recovery of the motion parameters of a sensor moving through a static environment, and more generally the relative motion between a sensor and a visible object. Under ideal conditions, once these motion parameters are known, a depth map can be recovered from two frames if the displacement (flow) field is exact.

In previous sections of this review, we have discussed various reasons why displacement fields are not perfect. Even with perfect information about sensor motion, displacement vectors from translational motion are a function of the depth of the surface element. Any ambiguity or error in displacements along linear paths emanating radially from the FOE leads to ambiguity in the depth of that surface element. There are several sources of such ambiguity including multiple minima in the matching process for computing displacements, noise affecting the match location, and finally the resolution in the matching process along that radial path. Consequently, we are viewing the matching process as a dynamic refinement of depth over multiple frames.

The work that we discuss here is a first step in the exploration of several issues involved in the stability, refinement, and prediction of depth maps over multiple frames [BHA85]. We are considering the differences in start-up (when no depth information exists) versus updating an existing (and possibly inaccurate) depth map; in both situations we assume limited computational resources are available, yet increasing accuracy over time is required.

When an image sequence is first acquired, or the visible field changes dramatically (as in the case of coming around a corner), no depth map exists and the situation can be considered as a start-up. Under an assumption of a fixed limit on the computation that can be carried out between any pair of frames, a strategy has been developed to extract a coarse depth approximation from the first pair of frames using a coarse spatial resolution for the matching process. Each subsequent frame that is processed can use the previous estimate of depth to narrow the match area while increasing the match resolution, thereby maintaining constant computation, but finer accuracy in the depth estimates. As this process continues, temporal resolution can also be reduced as necessary. Thus, the approach employed involves a combined hierarchical spatial and temporal resolution as frames continue to arrive.

The refinement strategy that we have just described for the start-up phase of depth map recovery can be generalised for updating, prediction, and error analysis. Under known sensor motion and known environmental depth, the image location and appearance of environmental features can be accurately predicted and matched from one frame to the next (leaving aside complex issues of image changes due to changes in lighting, highlights, shadows, shape distortion of surface patches, or occlusion). Thus, when one reaches the desired level (or limit) of spatial and temporal resolution, the updating process becomes one of prediction and verification of the environmental model. When predictions are not accurate, then depending upon the representation, the depth of either pixels, points, lines, regions, or surfaces could be refined in a focus-of-attention and refinement process for error reduction. Areas of the image and environment that do not behave as predicted become the focus of processing until their image dynamics over time can be properly predicted. In this manner one has an ongoing mechanism for verification of the current interpretation of the environment.

II. IMAGE INTERPRETATION

Work on the VISIONS system for interpretation of static images continues. A rule based system for generating initial object hypotheses from image data has been extended to permit information from multiple sources of low level data to be "fused" in a consistent manner. On the basis of the results in a forthcoming thesis by Weymouth [WEY85], we have refined the notion of schemas as a representation of knowledge. We are implementing a new schema system in Common LISP and translating existing schemas and their associated interpretation strategies into the new format. We are continuing to explore inferencing mechanisms based on the Shafer-Dempster-Lowrance idea of evidential reasoning [SHA76, DEM68, LOW82, WES83, WES85]. A recent development is a method for generating mass functions using explicit knowledge about the image domain without requiring that the range of values over which the mass functions are defined be either explicitly or implicitly discretised into "propositions".

II.1. RULE-BASED HYPOTHESES FROM COMPLEX AGGREGATIONS OF IMAGE EVENTS.

In a recent paper [WEY83] we described a simple type of knowledge source for generating object hypotheses for particular regions in the image. The rules are defined in terms of ranges over a scalar feature, and complex rules are defined as combinations of the output of a set of simple rules. The scores of these rules serve as a focus of attention mechanism for other, more complex knowledge-based processes. The rules can also be viewed as sets of partially redundant features each of which defines an area of feature space which represents a "vote" for an object on the basis of this single feature value. The region attributes include color, texture, shape, size, image location, and relative location to other objects. More recently, the approach has been extended to lines, with features including length, orientation, contrast, width, etc. In many cases, it is possible to define rules which provide evidence for and against the semantically relevant concepts representing the domain knowledge. While no single rule is totally reliable, the combined evidence from many such rules should imply the correct interpretation.

Most of the rules previously described are unary, accepting a region as input and returning a confidence for the object label. In addition, simple binary rules, defined over pairs of regions, were used to determine the similarity of the regions and to form aggregations of regions with similar properties. Typically, the rules operate on primitives formed by a single segmentation process (e.g. regions or lines) and result in the merging of the primitives into a more complete description, depending on the confidence returned by the rules. Forming more abstract groups of elements in this way has advantages when dealing with unreliable segmentation processes: fragmented elements can be grouped to form aggregates which perhaps more closely match object models.

Recently, we have extended this approach to include relational rules, which capture expected relations between the elements of multiple representation (e.g. regions, lines, surfaces) of the image data [BEL85]. Using rules of this form, sets of elements across the multiple representations can be selected and grouped on the basis of relational scalar measures associated with each rule. The result, assuming the confidence value returned by the rule is high enough, is the construction of complex aggregations of elements which satisfy user-specified relations across the multiple representations. One advantage of this approach is that it is modular and extensible; when new representations are added to the system, integration is accomplished by adding the appropriate rules.

In our preliminary work, we are concerned with relational rules defined over regions and lines. Since both are defined in a pixel-based representation, a convenient basis for the rules is intersection of the corresponding sets of pixels. Such relational rules, called intersection rules, are composed of three components:

1) a *filtering rule* for selecting lines which intersect a region based on relational measures;
2) a *ranking rule* which ranks the lines which intersect a region based on line attributes; and
3) a *combination function* which calculates the final score of the region-line aggregation based on the scores from the *filtering rule* and the *ranking rule*.

The relational measures are used to measure the type and degree of the relationship between a region and a line. Lines associated with regions are categorised into three types: boundary lines, interior lines, and lines which are neither interior nor boundary. The measures are:

1. interior-line-percentage: the ratio of line area interior to the region to total line area.
2. region-perimeter-percentage: the ratio of region boundary pixels covered by the line area to the region perimeter.
3. line-length-percentage: the ratio of the length of the region boundary covered by the line area to the total length of the line.

The filtering rule is then a complex line rule composed of a simple rule for each relational measure; in many cases it simply removes certain combinations of regions and lines from further consideration. The ranking rule ranks each line on the basis of how well it satisfies the associated relational measure. The combination rule is supplied the scores from the filtering rule, the ranking rule, and the relational measures and converts these into a confidence for the hypothesis supported by the rule.

These intersection rules can be used in some very diverse ways. One example is to use a filtering rule on interior-line-percentage to select only those lines which are interior to a region. The ranking rule could then be defined to

select short, high-contrast lines. The score of the ranking rule could then be averaged to form a complex texture measure. Alternatively, a density measure could be calculated by counting the occurrences of lines which receive a high score from the ranking rule and then normalising by the size of the region.

As an additional example, the length-percentage measure could be used to select lines which lie mostly on the boundary of the region. The ranking rule could then be defined to favor long lines. The scores from the ranking rule could then be averaged using region-perimeter- percentage as a weighting factor to form a simple shape measure.

A preliminary implementation of the extended rule system has been completed, several simple texture and shape rules have been written, and results have been obtained on urban house scenes and on road scenes. The results [BEL85] are quite promising. For example, we have been able to find roads in several roadscenes by using a rule which implements a simple shape measure. In the future, we intend to write additional rules and apply the system to a larger variety of images, develop new rule types, add additional representations for motion, depth, and surface segmentations, and incorporate [illegible]-based system into the schema system currently [illegible] (see next section.).

II.2. SCHEMA NETWORK REPRESENTATION [illegible]

In the VISIONS system [illegible] ne independent knowledge is represented in a hierar[illegible] schema structure organised as a semantic network [HAN[illegible], WEY83, PAR80, HAN83]. The hierarchy is structured to capture the decomposition of visual knowledge into successively more primitive entities, eventually expressed in symbolic terms similar to those used to represent the intermediate level description of a specific image obtained from the region, line, and surface segmentations. Each schema defines a highly structured collection of elements in a scene or object; each object in the scene schema, or part in the object schema, can have an associated schema which will further describe it. Each schema node has both a declarative component appropriate to the level of detail, describing the relations between the parts of the schema, and a procedural component describing image recognition methods as a set of hypothesis and verification strategies called *interpretation strategies*.

The schema system provides a hierarchy of memory structures, from vertices (or even pixels) at the bottom level through semantic objects at the top. A further division of knowledge into long term (LTM) and short term memory (STM) across the levels of hierarchy provides a convenient way of differentiating the system's permanent *a priori* knowledge base from the knowledge that it has received or derived from a specific image. The goal of the system is an interpretation, by which is meant a collection of objects at the top level of STM that is consistent with both the image date and the system's *a priori* knowledge of the world as represented in LTM.

A central problem of high-level vision is how to make use of knowledge, not just to categorise the results of lower levels of computation but also to guide those levels through the space of image analysis and feature extraction techniques. Practical systems will need to know about a extremely large number of objects – a prohibitive number for any system that attempts to find each object in each image. Furthermore, there is a computationally explosive number of low and mid-level image operations (segmentation algorithms, texture measures, line finders, rectangle finders, line grouping operators, etc. which collectively are termed 'knowledge sources') which might be applicable, especially when one realises that for almost every object there might be a variation of certain operations that would be particularly well suited to recognising *just that object*. As a result, the combinatorics of what low- and mid-level processes to apply and how to interpret their results is simply too great to expect any near-term increase in the power of computing systems to solve the problem by brute force computation. The high level vision system must heuristically control the work being done at the lower levels for computer vision to ever be computationally feasible. The goal of this research, then, is to provide a prototype knowledge driven system called the Schema System, to interpret images and provide control.

The development of the schema system confronts many of the same issues that have come up in other interpretation and control domains, such as speech understanding [LES75, WOO78]. Among them are questions of the knowledge representation, the communication of information, error recovery and the selection of knowledge sources.

A basic idea embedded in the schema system is that the knowledge of how to recognise an object is embedded in a *schema* of that object. In particular, the system has a set of *schema frames* which are procedures that comprise all of the system's knowledge that is unique to that object, including such pieces of information as what schema frame to start up to recognise a subpart and what knowledge sources would be particularly effective to recognise the object. In order to recognise an particular instance of an object in an image, the schema frame is *activated* to make a *schema instance*, which is a copy of the schema that is activated with a default set of parameters. The schema instance may, in turn, activate another schema frame to make another schema instance, and so on down the line. During the interpretation process, the set of active schema instances run concurrently and exchange information by writing and reading to/from a blackboard in order to create a single interpretation. The interpretation is a semantic network composed of different types of hypothesis nodes (one for each level of memory) and three kinds of links: links between hypotheses at one level, realisation links (between levels), and hypothesis links (between STM and LTM).

Inter-schema communication is accomplished via a blackboard. In general, there are three types of messages

that go on the schema blackboard: Hypotheses, Goals and Personal Mail. An important issue is when is information propagated, i.e. at what point does one schema instance's hypothesis effect another. We have adopted the basic principle that *the decision whether information should be propagated from one schema to another or not resides in the reader (given the blackboard communication), not the writer.* A schema instance must make sure the hypothesis has been posted by the time it is strong enough that another schema might use it.

Any system which manipulates uncertain information must confront the problem that its hypotheses will sometimes prove wrong and must therefore include mechanisms for error recovery. This is particularly a problem for blackboard style systems, where the failed hypothesis may have affected an unknown number of other decisions. One of the main features of this schema based approach is that all of the information about an object instance resides in one object hypothesis and the schema instance that created it. In particular, all of the dependancy information is already in that schema instance, along with the partial results used to calculate any decisions made as a result of a dependancy. The schema system therefore assigns to the schema that formed the hypothesis the duty of maintaining it. When the schema has finished everything else, it simply goes to sleep, waiting for its context to change. If the context does change, it wakes up, alters its hypothesis accordingly, and goes back to sleep. The result is that the schema system has true error recovery capability, avoids manipulating dependency lists, and can use previous partial results to calculate changes.

One obvious implication of the schema system as described is that it has no global monitor. Not only do schemas control knowledge sources, they control themselves in a distributed manner. There is no equivalent to the Hearsay Focus-of-control-database & scheduler [LES75] to decide what the system should do next. The advantage to this is that control in the schema system is more flexible. For example, it is easier to give a single object a unique relationship to its subparts than it would be in a centralised system, where the monitor might have to be altered in non-trivial ways. In fact, each schema instance may control its own resources in a different way. This means that more specialised knowledge can be incorporated into the control decisions, as opposed to being forced into extremely general methods. It also facilitates experimenting with control methods.

The first schema prototype being developed is called the OHM, or Object Hypothesis Maintenance schema. The name is to emphasize that the OHM does not simply create a hypothesis, it also maintains it as the interpretation process proceeds. Internally, the OHM is a collection of seven interpretation strategies (IS's), each of which runs as its own concurrent process. The most important IS is the OHM-control process. This strategy is responsible for deciding how the OHM and its hypothesis relates to the rest of the system. The remaining six IS's are Initial Hypotheses (typically inexpensive processes that give a first estimate as to whether the object exists in the image, and if so where), Hypothesis Expansions (e.g. an algorithm that expands a roof hypothesis, given just a corner of the roof), hypothesis support, conflict resolution, negative information (in general, how to use the information that something *isn't* a particular object), and information from subparts and/or superparts.

A programming shell has been created for research implementation of schema sets. Schema sets are groups of concurrent processes whose goal is to label a given type object, operating from high-level contextual and relation knowledge, and intermediate feature knowledge. The object labeling is implemented procedurally, which permits strategies to be tailored to the object being labelled with little interference from globally imposed data structuring. At the same time, the lack of a global controller imposes a great deal of structure on interprocess communication.

The purpose of the shell is to encourage development and testing of labeling strategies by optimising research and programming and testing time. A prototype shell is in place and 5 object schema types are at different stages of development and testing under the current shell. Feedback from these preliminary schemas will lead to improvements in the shell structure itself. The implementation is in Common LISP on the TI Explorers, with low level data and image processing functions handled on VAX.

II.3. INFERENCE NET

We are actively exploring the mathematical foundations of a knowledge representation framework within the domain of vision using the theory of evidential reasoning as developed by Dempster [DEM68] and Shafer [SHA76].

The Dempster-Shafer formalism for evidential reasoning supports an explicit representation of partial ignorance, uncertainty and conflict. The inferencing model allows "belief" or "confidence" in a proposition to be represented as a range within the [0,1] interval. The lower and upper bounds represent support and plausibility, respectively, of a proposition, while the width of the interval can be interpreted as ignorance.

The representation has two components [REY85]. The first part is static, and explicitly associates measureable properties of some feature of the image data, via knowledge sources, to labels which are to be assigned to abstractions of the image data. This association is made using the notion of a mass-function as defined by Shafer. These mass functions are generated using the notion of a possibility function which is defined using explicit knowledge about the image domain in question. Previous methods required that the range of values over which the mass functions are defined be either explicitly or implicitly discretised into "feature propositions".

The second part uses this static representation, a frame

of discernment, and the theory of evidence as developed by Shafer and by Lowrance to combine the mass functions (via Dempsters rule) and arrive at a consensus opinion for the purpose of determining the correct label of the image abstraction. Assumptions about the image domain are represented within the knowledge network via possibility functions; a conflict value detects when an assumption has been violated and is used as represention of uncertainty within the system.

Our representation provides a simple mechanism for representing uncertain information and for pooling of partial evidence. Assumptions one makes about the domain provide the constraints on the relationship between primitives extracted from the image data and objects in the scene one is trying to reason about; we are interested in obtaining and pooling evidence which pertains to these constraints. These include intrinsic properties of the objects, which are expressed as unary constraints, and contextual constraints such as spatial relationships which are binary or in general n-ary relations (for example adjacency is a binary relation, betweenness is a ternary relation).

III. INTERMEDIATE LEVEL VISION

The general strategy by which the VISIONS system operates is to build an intermediate symbolic representation of the image data using processes which initially do not make use of any knowledge of specific objects in the domain. The result is a representation of the image in terms of intermediate primitive such as regions, lines, and local surface patches with associated feature descriptors. These primitives may be directly associated with an object label (using the rule-based object hypothesis system as described in the previous section) or they may be grouped into more abstract descriptions. The grouping processes may be guided by high level contextual constraints (e.g. top-down) which effectively select certain groupings related to the interpretation goals, they may be guided by very general object-independent constraints (e.g. bottom-up), or they may be guided by both, changing their form depending on the constraints available.

In this section we summarize three areas of research whose focus is the construction of intermediate level primitives and their features.

III.1. GEOMETRIC GROUPING OF STRAIGHT LINES

The extraction of lines based on significant intensity changes and perceived boundaries between areas is a difficult and important step in image understanding. We have developed a new approach to the extraction of straight lines based on geometric grouping. The primary goal is the extraction of straight lines from images in which there are fragmented intensity discontinuities. The secondary goal is the demonstration that the use of geometric organization is an important part of the line extraction process and therefore can produce improvements when combined with standard edge detection techniques.

The algorithm has two major components: edge detection and hierarchical grouping. Hierarchical grouping has two steps which are performed at each level: linking and merging.

There are many edge detection algorithms which might be used. The main requirements are that it produce measurements of the intensity contrast and direction of the edge. The two algorithms which we have used for selecting points are zero crossings of the Laplacian operator [MAR80, CAN83] and the Haralick operator [HAR84].

The hierarchy is based on scale but there is no smoothing. The hierarchical representation has a number of advantages. It is a compact representation which reduces the search space at each level for sequences of linked edges. It reflects the observation that "closeness" of lines is scale dependent and is a multi-scale representation of a line which may be straight only at large scales.

The linking process is based on intrinsic and geometric properties. It searches a space of lines for almost collinear pairs which are close to each other and links the appropriate endpoints. There are four criteria used for linking:

1. Similar gradient magnitude. The gradient magnitudes across the edge must be close to each other and in the same direction.
2. The lines must be approximately collinear. Lines 180 degrees apart are not linked.
3. The end points of two candidate lines must be close.
4. The lines must not overlap. If both endpoints of one line project within corresponding endpoints of the other, they are not linked.

The merging process consists of grouping and replacement. If a sequence of linked lines can be approximated sufficiently well by a straight line, then they are grouped and are replaced by a straight line.

This approach has a number of advantages for extracting straight lines:

1. It links line segments even when they are separated by gaps.
2. Since it is based on gradient information and spatial information it can find lines which have low contrast as well as high.
3. Since it favors collinear line segments and uses gradient information, it is less sensitive to texture when extracting boundary lines. Zero crossings and algorithms which track edges are not able to distinguish between edges which are part of long, straight boundaries and lines of texture because they do not use geometric context.

The results shown in [WEI85] indicate that the principle of geometric grouping for extracting long straight lines gives significant improvement in the results obtained from standard edge detection algorithms. We plan to demon-

strate this over a wider range of images. This implementation is also a demonstration of the importance of geometric grouping in general; we plan to extend it to curved lines, (see Section III.3) parallel lines, closed contours, and other geometric abstractions.

Although the algorithm is very robust in its extraction of straight lines, it has some problems which we are continuing to investigate. The ability of the algorithm to bridge gaps is simultaneously one of its strengths and one of its weaknesses. Gaps sometimes appear in a line because of changes in the lighting conditions along the line, (such as shadows and specular reflections) which in turn affects the magnitude of the gradient. These gaps should be bridged. Other apparent gaps are caused by the alignment of distinct lines (such as those on the top or bottom of a pair of shutters); such gaps are real and should not be bridged yet at some level in the hierarchical representation they appear as one line. Methods must be found for analysing the multi-scale representation and for determining what scales are appropriate and which are not appropriate.

The algorithm, like many others, relies on intensity gradient information to link lines, yet what we perceive as straight lines are not always collections of edges with similar intensity gradients. Finally, the algorithm will often find long lines in heavily textured areas because of accidental alignment of texture edges. We are investigating the possibility of using texture measures to inhibit the linking step.

III.2. EXTRACTION OF CURVED LINES

Until recently the traditional method in computer vision for extacting straight and curved lines has been either through the use of the Hough transform or via "edge linking" algorithms applied to the output of some "significant edge pixel" algorithm. However, a novel approach for extacting straight lines was recently reported in Burns, Hanson and Riseman [BUR84], and involved a simple local computation (not involving any histogram methods) followed by a computation of connected components.

The central module of this algorithm was a grouping process using overlapping partitions on gradient orientation. In the context of extracting straight lines this process can be summarised as follows:

- Apply a gradient orientation measure to the image,
- Partition the output of this measure into overlapping sectors (normally 16 are used) and label the image according to the sector into which the gradient orientation falls,
- Apply a connected components algorithm to each non-overlapping partitioning,
- Apply a selection procedure to determine locally which partition is preferred (the edge support),
- Fit a straight line to the resulting "edge support" regions.

We have been investigating the application of this general approach to the problem of extracting semi-circular arcs, replacing gradient orientation with a curvature measure. Specifically we find "curve support regions" which are uniform with respect to a curvature measure and as such can be abstracted from the image data as a part of a circle.

The curvature measure we are using is given by the Kitchen-Rosenfeld curvature operator [KIT80] defined by:

$$K = I_{xx} \cdot I_x^2 + I_{yy} \cdot I_y^2 - 2I_{xy} \cdot I_x \cdot I_y (I_x^2 + I_y^2)^{3/2}.$$

In fact this measure only makes sense when applied to areas of locally maximum gradient magnitude, ie. zero crossings of some second derivative operator. Thus our algorithm can be summarised as follows:

- Apply the curvature measure along the zero-crossing contour.
- Partition the range of the curvature measure into overlapping sectors.
- For each partition, label each pixel according to the partition into which the curvature value falls.
- Produce regions by applying a connected components algorithm to the labels of the curvature partitions.
- Fit a semi-circle to each curve-support region.
- Each pixel then votes for the region whose extracted curve is longest. The percentage of pixels within a region that vote for that region is the support of the region.
- Normally the regions selected are those whose support is greater than 50 percent.

Associated with each curve is a set of curve attributes, such as length, center, radius, endpoint parameters, contrast and support. The algorithm is local in nature and is robust in the face of moderate amounts of noise due to the coarse partitioning of the output curvature measure .

In summary, we have developed a system to derive local and piecewise circular descriptors of the image data. The approach utilises local 2D operations and is computable in parallel. Curves are partitioned based on constancy of curvature rather than usual extrema methods. Also in contrast to other approaches, descriptors of neighboring segments are treated as independent, with the expectation that higher level processes will guide the next level of grouping. The system is designed to provide reliable local primitives (as opposed to pixel level events) for the purpose of moving up the abstraction hierarchy within the image understanding system.

III.3. APPLICATION OF VANISHING POINTS TO 3-D MEASUREMENT

Perspective is an important cue to 3d spatial information such as the direction of lines or the orientation of surfaces. Human beings can perceive three-dimensional objects in space even when looking at two-dimensional images. A computer vision system must do likewise, but 3D shape, size and location cannot be recovered from a single image without additional information. Vanishing points and vanishing lines can provide this information in the case of objects which have parallel lines or edges on planar surfaces. Once the location of the vanishing point is detected, we can use it as a cue to calculate the distance and shape of the object to which the parallel lines belong [NAK80, NAK84a, BAR82].

Estimation of the errors in these features has practical significance and could be used in many ways. With some object models such as buildings, the dihedral angles between their surfaces (e.g. walls and roofs) are invariant shape features. An estimate for the relative orientation between the surfaces which incorporates the error allows one to verify hypotheses about image which correspond to planar surfaces. For example when analysing a house scene, if two adjacent regions are temporarily labeled as house walls based on some property, say rectangular shape, the calculation of the mutual angle of the two surfaces can be used to verify this. If the angle is calculated to be 90 degrees, it will support the labelling as walls. However, we also want to know when we should reject the hypothesis. This means that we must know whether the measured angle is outside the estimated range of error. A modular process such as a knowledge source on perspective could use this type of information in the form of constraints to generate and verify hypotheses. In a bottom-up approach, the range of error could be used to limit the search space when looking for planes which are perpendicular.

Parallel lines in 3d space are projected onto the image plane to lines which radiate from a single common point, called a vanishing point (VP). It can be used to calculate the size and orientation of objects with parallel lines. The vanishing line for a surface can be computed as the line passing through two VP's obtained from two sets of parallel lines. There is an infinitely many sets of parallel lines which could be drawn in a given plane and the vanishing point for each set lies on this vanishing line.

The surface orientation of a plane is given by the unit normal vector perpendicular to the surface. The vanishing line (VL) of a plane gives a precise description of the unit normal. The distance from the VL to the center of the image plane corresponds to the angle of tilt of the surface away from the viewer. If the line goes through the center, then the normal to the surface is parallel to the viewing plane. The second angle of the surface is given by the orientation of the vanishing line; its normal is the projection of the normal to the surface onto the image plane. Thus, we analyse how the errors in the distance and orientation of the vanishing line affect the estimates for the surface orientation.

We have done an analysis of how errors in the location of these vanishing points affects our estimate of surface orientation and line length (NAK84b). We have developed formulae for these errors as a function of VP or VL errors, and we have used constraints based on real world knowledge to increase the precision of the estimates of surface orientation.

The assumptions made are that the focal length of the camera and the depth of one point on a line are known. If the depth is not known at all, then the orientation can still be recovered, but only relative distances can be estimated.

The algorithm for locating vanishing points is a form of Hough transform:

1. Extract lines which are likely to be parallel
2. Project those lines stereographically onto half of the Gaussian sphere and extend them to semicircles
3. Locate peaks by thresholding

The estimation of the surface normal from the estimates for two vanishing points involves intersecting constrained regions on the Gaussian sphere:

1. Each vanishing point estimate, which is an area on the Gaussian sphere, determines an annular set of possible directions for the normal to the surface.
2. The intersection of these annular sets is the estimate for the normal.

For the application of geometric constraints in the case where two house walls are perpendicular, the estimate for the normal for one wall was rotated 90 degrees on the Gaussian sphere and intersected with the estimate for the normal to the other wall. In our experiments, there was significant reduction in the size of the region estimate for the surface normal in the example used.

Although we assumed the perpendicularity between the planes in the two cases mentioned above, we can apply this method even if the dihedral angle is not a right angle. If the angle is given to be θ_S, the other plane's normal must be in the belt that makes an angle θ_S with the given normal. We can again form the consistent range by taking the intersection of the belt and the area for the VP. These constraints can also be applied to more than two planes, for example when three planes meet in a trihedral angle.

IV. THE UMASS IMAGE UNDERSTANDING ARCHITECTURE PROJECT

UMass is designing and constructing a highly parallel architecture for computer vision with the goal of achieving real-time processing rates for low, intermediate and high

level image interpretation tasks. This architecture consists of three tightly coupled layers that correspond to these levels of abstraction. These layers are the Content Addressable Array Parallel Processor (CAAPP) at the bottom, Intermediate and Communications Associate Processor (ICAP) in the middle, and the Symbolic Processing Array (SPA) on top. Attached to the SPA is a host processor.

The CAAPP is an associative square grid processing array that is designed to provide bi-directional parallel communication between symbolic sensory processing [WEE83, WEE84, LEV84]. The ICAP is also an associative square array, and is tightly coupled to the CAAPP and SPA. The purpose of the ICAP is to perform intermediate level symbolic processing and to facilitate the flow of information and control between the CAAPP and SPA. The SPA is an array of processors which perform high level symbolic processing such as hypothesis generation and testing, schema processing, and knowledge source/blackboard processing.

The multilayer associative structure of the UMass architecture provides simultaneous parallelism at three different levels of abstraction with high bandwidth bi-directional flow of information and control between the levels. This permits the entire iconic to symbolic transformation process to take place within the architecture so that the top layer can provide a high level symbolic interface to the image interpretation process. At this level, images of the environment have essentially been transformed into a symbolic representation of that environment.

The effort involves a custom VLSI implementation for the processing elements in the bottom two layers of the architecture; a systems hardware implementation for integrating the custom processors with off-the-shelf components in the top layer and host processor; a software development for creating a complete programming environment, simulators and tools for the system; and an algorithms effort for implementing vision algorithms on the architecture. Much of the hardware effort and part of the software and algorithms efforts will be shared with Hughes Research Labs.

IV.1. Hardware:

A test chip of the NMOS version of the CAAPP processing element has just been received from the MOSIS facility. We are currently preparing to test this chip.

The layout for a CMOS version of the CAAPP processing element is about 60 percent complete. We will be examining the tradeoffs involved in going to a CMOS implementation. Although CMOS would increase the size of the layout, it would permit the use of the MOSIS scaleable rules, with a potential for significant size reduction and speed increase in the future.

The first pass on the design for the Intermediate and Communications Associative Processor (ICAP) has been completed. Unfortunately, to place this ICAP design on the same chip as the CAAPP cells will necessitate a greater number of pins than is currently available from MOSIS. We are thus examining the tradeoffs of reducing the functionality of the ICAP to make it fit the pin limitations, versus placing the ICAP on a separate chip. The latter would double the size of the prototype circuit boards, but would provide greater processing flexibility.

IV.2. Software and Algrithms

We are currently constructing an instruction-level functional simulator for the new CAAPP architecture. This simulator promises to provide considerably greater execution speed than the old simulator. The new simulator is being written in C as a stand-alone, portable system although its image formats will be compatible with the UMass VISIONS system. Once the ICAP architecture is finalised, it will also be incorporated into the simulator.

An iconic to symbolic transformation process has been developed and tested for the CAAPP, using a version of the VISIONS system to simulate the new CAAPP architecture, prior to construction of the new simulator. Several vision algorithms have been implemented in the simulator; these include an algorithm for computing approximations to large Gaussian convolutions, the Burns' line extraction algorithm, and the line grouping algorithm described in Section III.1.

A prototype slice of the UMass architecture is scheduled for completion in approximately 2 years. This will produce a symbolic representation of region and line image events, as well as surfaces, and can be interfaced to a LISP processor as a demonstration of the concept. The complete prototype could be built in two additional years. At the end of the first year the software effort will produce simulators and tools for the bottom two layers of the architecture. The second year of the software effort will result in a transportable, stand-alone simulator for the entire architecture with associated environment and tools. After this, the software effort will concentrate on implementing vision processing tasks on the simulators and then transferring those implementations to the hardware as it becomes available. Additional enhancements to the environment and further tools will be developed as necessary.

REFERENCES

[ADI85a] G. Adiv, "Determining 3-D Motion and Structure from Optical Flow Generated by Several Moving Objects," *IEEE Trans. Pattern Anal. Machine Intell.*, Volume PAMI-7, July 1985, pp. 384–401.

[ADI85b] G. Adiv, "Interpreting Optical Flow," Ph.D. Dissertation, Computer and Information Science Department, University of Massachusetts at Amherst, September 1985.

[ANA84] P. Anandan, "Computing Dense Displacement Fields with Confidence Measures in Scenes Containing Occlusion," *SPIE Intelligent Robots and Computer Vision Conference*, Volume 521, 1984, pp. 184–194; also *DARPA IU Workshop Proceedings*, 1984; and COINS Technical Report 84-32, University of Massachusetts at Amherst, December 1984.

[ANA85] P. Anandan and R. Weiss, "Introducing a Smoothness Constraint in a Matching Approach for the Computation of Optical Flow Fields," *Proc. of the Third Workshop on Computer Vision: Representation and Control*, October 1985, pp. 186-196; also in *DARPA IU Workshop Proceedings*, 1985.

[BAR82] S.T. Barnard, "Interpreting perspective images," SRI Technical Note 271, 1982.

[BEL85] R. Belknap, E. Riseman, and A. Hanson, The Information Fusion Problem and Rule-Based Hypotheses Applied To Complex Aggregations of Image Events, Proc. DARPA IU Workshop, Miami Beach, FL December 1985.

[BHA85] R. Bharwani, A. Hanson, E. Riseman, Refinement of Environmental Depth Maps over Multiple Frames, Proc. DARPA IU Worksop, Miami Beach, FL. December 1985.

[BUR84] J.B. Burns, A. Hanson, and E. Riseman, Extracting Linear Features, Proc. 7th ICPR, Montreal, 1984. Also COINS Technical Report 84-29, August 1984. To appear in IEEE PAMI.

[CAN83] J.F. Canny, Finding Edges and Lines in Images, MIT AI Lab Technical Report No. 720, June 1983.

[DEM68] A.P. Dempster, A Generalisation of Bayesian Inference, Journal of the Royal Statistical Society, Series B, Vol. 30, 1968, pp. 205-247.

[GLA83] F. Glaser, G. Reynolds, and P. Anandan, Scene Matching by Hierarchical Correlation, Proc. IEEE CVPR, June 1983, pp. 432-440.

[HAN78] A. Hanson and E. Riseman, *VISIONS: A Computer System for Interpreting Scenes*, Computer Vision Systems (A. Hanson and E. Riseman, eds.) (1978), 303 - 333, Academic Press.

[HAN83] A. Hanson and E. Riseman, *A Summary of Image Understanding Research at the University of Massachusetts*, COINS Technical Report 83-35 (October 1983), University of Massachusetts at Amherst.

[HAR84] R.M. Haralick, Digital Step Edges from Zero Crossing of Second Directional Derivatives, IEEE Trans PAM 6, January 1984, pp. 58-68.

[HOR81] B.K.P. Horn and B.A. Schunck, "Determining Optical Flow," *Artificial Intelligence*, Volume 17, 1981, pp. 185-203.

[KIT80] L. Kitchen and A. Rosenfeld, A Gray Level Corner Detector, Tech. Report No. 887, Computer Science Center, University of Maryland, College Park, MD, 1980.

[LAW84] D.T. Lawton, Processing Dynamic Image Sequences form a Moving Sensor, Ph.D. Dissertation (TR 84-05), Computer and Information Science Department, University of Massachusetts, 1984.

[LES75] V.R. Lesser, R.D. Fennell, L.D. Erman, and D.R. Reddy, Organisation of the Hearsay-II Speech Understanding System, IEEE Trans. on ASSP 23, pp. 11-23.

[LEV84] S.P. Levitan, Parallel Algorithms and Architectures: A Programmers Perspective, Ph.D. Dissertation (COINS Technical Report 84-11), Computer and Information Science Department, University of Massachusetts, May 1984.

[LOW82] J. Lowrance, Dependency Graph Models of Evidential Support Ph.D. Thesis, University of Massachusetts, Amherst, 1982; also COINS Technical Report No. 82-26.

[MAR80] D. Marr, and E. Hildreth, Theory of Edge Detection, Proc. of the Royal Society of London, B., 207, pp. 187-217.

[NAK80] H. Nakatani, et al. "Extraction of vanishing point and its application to scene analysis based on image sequence," 5th Int. Conf. on Pattern Recognition, pp. 370-372, 1980.

[NAK84a] H. Nakatani, T. Kitahashi, "Inferring 3-d shape from line drawings using vanishing points," 1st Intn'l Conf. on Computers and Applications, 1984.

[NAK84b] H. Nakatani, R. Weiss, and E. Riseman, "Application of Vanishing Points to 3D Measurement," Proc. SPIE, Vol. 507, 1984, pp. 164-169.

[PAR80] C.C. Parma, A.R. Hanson and E.M. Riseman, *Experiments in Schema-Driven Interpretation of a Natural Scene*, COINS Technical Report 80-10 (April 1980), University of Massachusetts at Amherst.

[PAV85] I. Pavlin, A. Hanson, and E. Riseman, Analysis of an Algorithm for Detection of Translational Motion, Proc. DARPA IU Workshop, Miami Beach, FL, December 1985.

[REY85] G. Reynolds, D. Strahman, N. Lehrer, Converting Feature Values to Evidence, Proc. DARPA IU Workshop, Miami Beach, FL, 1985.

[SHA76] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, 1976.

[STR84] T. Strat, Continuous Belief Functions for Evidential Reasoning, Proc. AAAI-84, pp. 303-313.

[ULL79] S. Ullman, The Interpretation of Visual Motion, MIT Press, Cambridge, MA 1979.

[WEE83] C. Weems, S. Levitan, D. Lawton, and C. Foster, A Content Addressable Array Parallel Processor and Some Applications, Proc. DARPA IU Workshop, Arlington, VA, June 1983.

[WEE84] C. Weems, S. Levitan, C. Foster, E. Riseman, D. Lawton, A. Hanson, Development and Construction of a Content Addressable Array Parallel Processor (CAAPP) for Knowledge-Based Image Interpretation, Proc. Workshop on Algorithm-Guided Parallel Architectures for Automatic Target Recognition, Leesburg, VA July 16-18, 1984, pp. 329-359.

[WEI85] R. Weiss, A. Hanson, and E. Riseman, Geometric Grouping of Straight Lines, Proc. 1985, DARPA IU Workshop, Miami Beach, FL, 1985.

[WES82] L. Wesley and A. Hanson, The Use of an Evidential-Based Model for Representing Knowledge and Reasoning about Images in the VISIONS System, Proc. Workshop on Computer Vision, Rindge, NH, August 23-25, 1982.

[WES83] L. Wesley, Reasoning about Control: The Investigation of an Evidential Approach, Proc. 8th IJCAI, Karlsruhe, West Germany, August 1983, pp. 203-210.

[WES85] L. Wesley, Ph.D. Thesis, University of Massachusetts, Amherst, in preparation.

[WEY83] T.F. Weymouth, J.S. Griffith, A.R. Hanson and E.M. Riseman, *Rule Based Strategies for Image Interpretation*, Proc. of AAAAI-83 (August 1983), 429-432, Washington D.C. A longer version of this paper appears in Proc. of the DARPA Image Understanding Workshop (June 1983), 193-202, Arlington, VA.

[WEY85] T.E. Weymouth, Using Object Descriptions in a Schema Network For Machine Vision, Ph.D. Dissertation (in progress), Computer and Information Science Department, University of Massachusetts, Amherst.

[WOO78] W.A. Woods, Theory Formation and Control in a Speech Understanding System with Extrapolation Towards Vision, in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, 1978.

SECTION II

INVITED TECHNICAL REPORTS

GC970145-10

KNOWLEDGE-BASED INTERPRETATION AIDS TO THE NAVY OCEANOGRAPHIC IMAGE ANALYST

ICDR J. D. McKendrick
Matthew Lybanon
Naval Ocean Research and Development Activity
Code 321
NSTL, Mississippi 39529-5004

ABSTRACT

Satellite imagery of the oceans is adding synoptic coverage to in situ measurements of traditional oceanography. Except for SEASAT and the Navy GEOSAT, almost all remote measurements have been made from weather satellites. In a few years, the Navy N-ROSS satellite is to place in orbit four microwave instruments designed to make oceanographic measurements. The deluge of oceanographic data provided by N-ROSS and other satellites will pose serious problems for operational interpreters, which will be added to those caused by the impracticality of automated interpretation and the uneven quality of human interpretation. Expert systems that incorporate a knowledge base and inference mechanism offer a possible solution to the dilemma. Expert systems may be able to advise inexperienced interpreters and, eventually, may lead to automation of the advisement process. A study of the problem area recommended the implementation of a prototype expert system that knows about mesoscale ocean features. That system could be used to organize knowledge about oceanographic image understanding, which, in turn, could be used to develop a more powerful expert system.

INTRODUCTION

Traditional oceanography has relied on in situ instruments – thermometers, salinometers, conductivity meters, etc.—to measure oceanic parameters. While those measurements typically have provided data as a function of depth, it has been difficult to make horizontal measurements at a close enough spatial and temporal spacing to reveal time varying mesoscale features. Satellite observations have added that new horizontal dimension. Those observations, in particular, give information about the surface layer of the ocean, while their synoptic coverage often provides key indicators concerning oceanographic mesoscale changes. Satellite imagery allows for a more complete understanding and large scale context for the incorporation of isolated, in situ, measurements.

A great deal of satellite oceanography has used infrared (IR) and visible imagery from meteorological satellites. While the sensors in those satellites were not designed for oceanographic measurements, and despite problems such as cloud cover and signal attenuation by the atmosphere, oceanographers have made good use of the observations. A few satellites have been designed specifically for oceanography. Among these are SEASAT, which, unfortunately, failed about three months after launch in 1978 and the U. S. Navy GEOSAT, which placed a microwave altimeter in orbit in March 1985.

The Navy Remote Ocean Sensing System (N-ROSS) is to be another dedicated ocean sensing satellite and is scheduled for a 1990 launch. It will carry four microwave instruments to make all-weather ocean observations. Those instruments are an altimeter, a low-frequency microwave radiometer (LFMR), a scatterometer (NSCAT) , and a special sensor microwave/imager (SSM/I). The primary ocean parameters to be measured by each sensor include

- altimeter—sea surface dynamic topography (mesoscale features),
- LFMR—sea surface temperature,
- NSCAT—surface windspeed and direction,
- SSM/I—windspeed and sea ice.

(Note: The altimeter and NSCAT are nonscanning/nonimaging, but the LFMR and SSM/I are scanning/imaging.) The stream of data from N-ROSS, when added to that from other satellites (including other planned new satellites), will inundate the oceanographic remote sensing analyst with data.

At present, the Navy must use a human/machine mix to exploit satellite data to provide oceanographic information to the Fleet. But, human interpretation is of uneven quality and is labor-intensive. Because of the subjective nature of the human interpretative process and the varying skill levels of the interpreters, it clearly makes sense to strive to standardize and to optimize the interpretation function. Then the quality of the products would be less sensitive to inexperience, fatigue, and similar factors. It would be useful to transfer existing laboratory expertise in image enhancement techniques and other machine aids to the operational interpreter in the field. This would be helpful to operational centers, which must meet operational schedules, sometimes with inexperienced personnel.

Conventional automated techniques for satellite data interpretation (e.g., standard signal processing techniques, image segmentation and classification) do not significantly lend themselves to satellite oceanography for several reasons. Among these are: ocean features are time varying, no efficient mathematical characterization of the features generally exists, and images are frequently cloud-covered or otherwise contaminated with noise [1]. An example of the latter problem is that the surface thermal signature (in IR imagery) of a cold-core eddy may be masked by solar heating of the surface layer or obscured by a humid marine boundary layer.

However, at NORDA we have found that interpretation by human experts using interactive image processing techniques frequently works well despite these problems. It also appears that an automated approach that uses methods similar to those of human experts would overcome both the problems of conventional automated analysis and the problems of human interpretation discussed above. This suggests that the prospect for an oceanographic image analyst expert system

offers a potential solution to the problem of operationally obtaining oceanographic information from satellite observations.

Some potential payoffs of the application of expert systems technology to the above problems are the expectation of raising the performance of less experienced analysts to expert level, and of the longer-term possibility that a computer program may ultimately be able to replace the human analyst. The latter goal, if achieved, would replace the current subjective nature of ocean image interpretation and make it objective.

Figure 1 is a high resolution (1.1 km x 1.1 km pixel) NOAA-7 IR (channel 4) image of a portion of the Gulf Stream area southeast from Nantucket. Viewing conditions were particularly clear, and following a linear contrast enhancement the image has been further enhanced by a modified Chen-Frei edge enhancement filter [2]. Thanks to the enhancement, the strong temperature gradient between the Gulf Stream (to the south) and colder slope water mass (to the north), as well as the temperature signatures of several eddies and smaller vortices are easily seen. The image graphically depicts aspects of the synoptic oceanography of the region. It is likely that conventional image understanding techniques (edge detection, segmentation, classification) could provide adequate analysis of this image. Even so, an experienced human interpreter would have the advantage of knowing such shortcuts such as "the approximately circular features south of the Gulf Stream are probably mesoscale features called eddies, and their large central cores of slowly counter-clockwise rotating water are colder than the surrounding water mass." While existing automated image understanding might do the job, human interpretation does it better.

Figure 2 is more typical of satellite IR images of the ocean. While it is possible to recognize some of the ocean features in the scene, portions of the image are cloud-covered. Traditional image processing techniques would experience failures at trying to determine the oceanographic context of the scene. An experienced interpreter—who is particularly knowledgable of the region—would have a much easier time.

Figure 1. Edge enhanced very clear satellite IR Gulf Stream image.

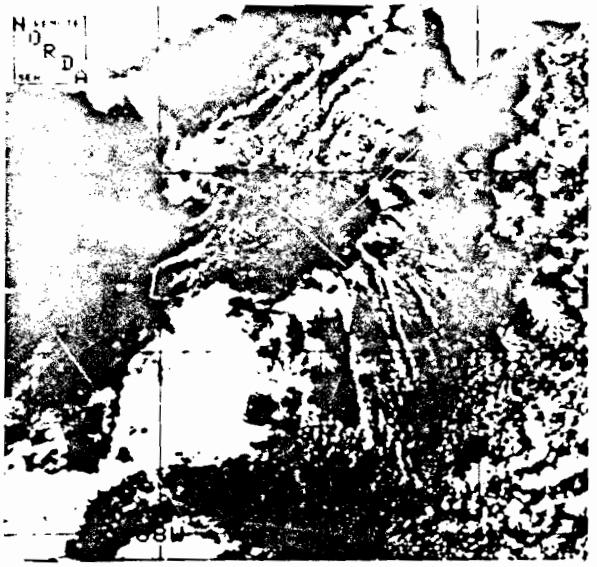


Figure 2. Cloudy satellite IR image of same area.

Considering the above sketch of the problem, it is possible to outline some of the characteristics of a hypothetical computerized system to support the operational oceanographic image analyst. Rather than performing large scale numerical calculations to solve problems of a predefined type in a step-by-step manner, the system should be able to employ past experience to solve new problems. It should be able to draw conclusions from a store of task-specific knowledge. It should be able to draw those conclusions principally through logical or plausible inference, not necessarily by numerical calculation. It should be able to do the type of things which a human expert does after "years of experience."

NORDA SATELLITE IMAGERY PROCESSING FACILITY

The following description presents the context of NORDA's capabilities for working with imagery data. The NORDA Satellite Data Receiving and Processing Facility (SDRPS) offers the latest in capabilities for acquiring digital satellite ocean imagery whose scenes may cover any part of the globe [3]. Data signals can now routinely be received from the following satellites: NOAA-Series polar orbiters, Geostationary Operational Environmental Satellites (GOES), and the Defense Meteorological Satellite Program (DMSP). Imagery is, however, only collected on request to support ongoing research projects.

The Interactive Digital Satellite Imagery Processing System (IDSIPS) consists of three I²S Model 70 image processing work stations and provides the NORDA scientists with the main image processing capability. These systems are used for research and the development of applications algorithms.

The research and development being conducted for the GEOSAT Program has a separate I²S Model 75, interfaced with a Gould SEL 32/27. A classified data transmission line from JHU/APL brings satellite altimeter data into the GEOSAT processing facility. From the GEOSAT altimeter data and following considerable processing, the dynamic topography (subtrack) of the ocean surface is resolved.

A broad range of oceanography related interactive image analysis projects are being conducted by the Remote Sensing Branch. They

include basic research at the 6.1 level through advanced development 6.3 level.

PLANNED WORK

Interpretation of satellite imagery requires familiarity with several disciplines: the physics and geometry of satellite remote sensing; characteristics of features of interest (features in the region observed and the general oceanographic context); and data manipulation, analysis, and display techniques. It seems likely that a successful oceanographic data processing/interpretation expert system could assist experienced researchers as well as inexperienced workers.

A study for NORDA [4] that considered potential applications of Expert Systems techniques to the problem of understanding satellite ocean images recommended the the development of a system to support the interpretation and understanding of mesoscale ocean features as the application most likely to produce results in the near future. The anticipated large increase in the volume of remotely sensed data was cited, but another basis for the recommendation was the recognition that the demand for more detailed, more specialized synopses is likely to increase. Consequently, since processing the relevant information to provide timely synopses at the desired level of detail will require a high degree of automation, coordination, and expertise, this area was selected as a candidate for expert systems development.

The nature of the problem is sufficiently complex that development is likely to be required in areas other than expert systems per se. Three other areas are involved:

- Image sequence analysis—the processing of multi-temporal image sequences of a given area in order to better understand the time and space evolution of ocean features of interest. Not only must individual images be processed, but additional information of a sequential nature must be extracted (such information may make the processing of new images simpler or more reliable).
- Multi-sensor data integration—a requirement to ensure that inferences about features are based on consistent processing of all data, imagery and other types. The combination of detailed along-track coverage provided by altimetry and the synoptic view provided by IR imagery, as recommended by Leitao, Huang, and Parra [5] and as is now being used at NORDA in the Navy GEOSAT Ocean Applications Program [6], is an example.
- Image understanding-the processing of individual images to obtain a decomposition into ocean features as well as the processing of image sequences to develop a representation of the evolution of those features.

The system recommended for development is one intended to support the analysis of evolutionary, structural changes in features, such as ocean fronts and eddies in an area of the ocean, by processing a series of registered images and other sensory information from that area. In that prototype interactive, semiautomated processing is to be supported by an expert system. Figure 3 shows the proposed organization of such a system. The data base consists of two parts: a static data base (SDB) and a dynamic data base (DDB). The SDB would contain the knowledge base (facts and inference rules) that would represent the then current understanding of the problem domain. That knowledge base would likely be composed of subjective knowledge of experts plus quantitative or statistical results of studies, such as studies of the space-time structure and variability of specific water masses. The DDB would contain the then current facts about the area under investigation, plus a model of the area that would

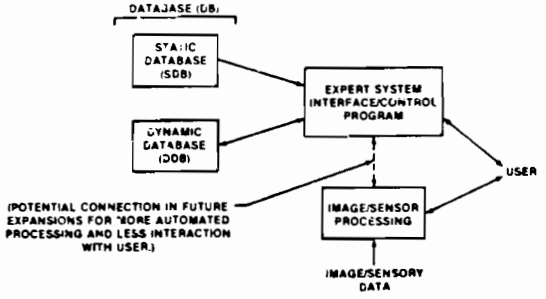


Figure 3. Organization of prototype.

describe suspected features, boundaries, etc., and information on the evolution of dynamic processes. The expert system would contain the "inference engine" to apply inference rules (from the SDB) to information in the DDB; the results of the application rules would be new facts in the DDB. In the initial system there would be no direct connection between the image processing workstation and the expert system. A later version is planned in which the two would be coupled.

The initial system will provide a prototype for experimentation and development of techniques. The experimentation will reveal flaws in the initial set of inference rules, and will help in the organization of oceanographic knowledge into a more efficient form for system implementation. The acquisition of new knowledge in ocean image interpretation and its subsequent use in the system will make it possible to enhance the prototype. By a "bootstrap" approach, it will be possible to develop progressively more refined, more sophisticated system. That later system should make possible more automated, less interactive image/sensor data processing. The development of this prototype system will accomplish some of the longer range goals set forth earlier in this paper.

REFERENCES

1. Lybanon, M. and J. D. McKendrick, "Some Applications of Image Processing in Oceanography," 15th South Eastern Symposium on System Theory, March 1983.

2. Frei, W. and C. Chen, "Fast Boundary Detection: A Generalization and a New Algorithm," *IEEE Transactions on Computers*, V C 26, No. 10, October 1977.

3. Hawkins, J. et al. "Remote Sensing at NORDA," *Eos Transactions*, American Geophysical Union, Vol. 66, No. 23, pp. 482–[illegible] June 4, 1985.

4. Thomason, M. G. and J. R. B. Cockett, *Expert System Support for the Interpretation Understanding of Oceanic Images and Sensory Data*, Final Report Contract PON 0001484M0025, May 31, 1984.

5. Leitao, C. D., N. E. Huang, and C. G. Parra, "A Note on the Comparison of Radar Altimetry With IR and In Situ Data for the Detection of the Gulf Stream Surface Boundaries," *Journal of Geophysical Res.*, Vol. 84, No. 88, pp. 3969–3973, 1979.

6. Lybanon, M. *GEOSAT Ocean Applications Program (GOAP) Initial Data Processing and Analysis System Test and Evaluation Plan*. NORDA Technical Note 270, Naval Ocean Research and Development Activity, NSTL, Mississippi, April [illegible]

GC970145-11

The ESPI Vision System

Thomas C. Rearick

Lockheed-Georgia Company
Marietta, Georgia 30063

Abstract

The ESPI vision system is an experiment to determine the feasibility of achieving very fast domain-independent computer image understanding. The term "understanding" is meant to imply the construction of some conceptual representation that is consistent with both internalized world knowledge and externally sensed data. Data reduction and modular design are discussed with respect to satisfying real-time design criteria. A new form of image representation is demonstrated which features bandwidth reduction, lossless reversibility (to and from pixel representation), and ease of computer manipulation and exploitation of global data dependencies. The use of pre-attentive vision and focus of-attention operators are shown to reduce processing bandwidth and the incidence of backchaining. A novel approach to the implementation of real-time hardware is proposed.

I. Introduction

The objective of the Experimental Symbolic Processing of Imagery (ESPI) effort is to rapidly prototype a system that demonstrates the feasibility of real-time image understanding. In order to achieve this goal, the ESPI vision system implements several novel and original features. This paper discusses some of these features as well as their philosophical justification. Rapid prototypes can identify high risk problem areas quickly so that the scope and dimension of a problem may be assessed.

The term "image understanding", as used in this paper, is defined as the construction of conceptual representations or models which are consistent with both internalized world knowledge and one or more two-dimensional image patterns being sensed. This definition of "image understanding" is general enough that human perception may be included. This definition does rule out many classical pattern recognition systems, however. One would have to stretch the definitions of conceptual models and world knowledge to include feature spaces and discrimination functions. Conceptual models might include frames, augmented transition networks, semantic nets, conceptual dependency notations, or semantic grammars. While feature spaces assume continuous and probabilistic models, conceptual models do not. One of the best examples of "understanding" is MARGIE [Sha73] and the series of programs from Yale that have followed it. These programs, based on Shank's Conceptual Dependency [Sha72] meaning representation language, create representations that are consistent with a natural language input and internalized knowledge (in the form of scripts, plans, and goals).

This definition of image understanding rules out parametric vision systems which attempt to construct three dimensional geometric or volumetric models. Examples of these approaches include photometric stereo or structured light. Volumetric models are not conceptual structures, though they may use propagation of constraints in order to achieve consistency. Examples of early image understanding systems (in the broader sense) include the work of Roberts [Rob65], Guzman [Guz68], and Huffman and Clowes [Clo71]. More recent image understanding systems include the photo interpretation system of Nagao and Matsuyama [Nag80], VISIONS [Han78], ACRONYM [Bro79], and Mapsee2 [Hav80].

This paper does not attempt to describe the architecture, the application, or the implementation of the ESPI vision system. A few novel features, research objectives, and preliminary results are presented. Section II states explicitly the methodological perspective used in this research [Hal85]. Section III discusses three

design principles which are common to the ESPI vision system as well as to all biological vision systems. Section IV presents three aims of active research and results to date.

II Methodological Perspective

The aim of this research is to seek general principles of intelligence and image understanding which apply to natural as well as artificial systems. The ESPI vision system is an attempt to develop one instance of an artificial domain independent image understanding system which lends support to general theories about all image understanding systems (animal, insect, or machine). Sources of these general theories include the physical sciences and, to a lesser extent, the biological sciences. Biological processes and mechanisms are considered useful for supporting these general principles only when their physical mechanisms are understood and when their purpose is understood within the context of evolution, ecological niche, and cultural interaction.

III. Philosophical Basis

The development of the ESPI vision system is based on a few pragmatic design guidelines as well as general vision principles. A few of these guidelines and principles are discussed separately although they are related.

All Solutions Are Not Real-time

Response time constraints are an integral part of the image understanding problem. Image understanding systems will be of little use in future applications if they are unable to operate in "real-time". There may be many solutions to the general vision problem but very few of those may have practical implementations. To use a game playing analogy, it is understood how to write a program that selects the best move in chess (assume that it is sufficient to calculate all possible moves to thirty levels). We don't know how to implement this program so that we may be assured of getting an answer in our lifetime. Humans provide an example of what is possible in complex tasks such as chess playing or image understanding if one is willing to trade off guaranteed optimality for speed.

Development of the ESPI vision system is being guided by several real-time design guidelines. These include 1) nondispersive or constant-time search, 2) rigidly controlled backward chaining, 3) no non-deterministic control mechanisms, 4) no synchronized feedback, and 5) data and width reduction. This last design criteria is discussed next.

Reduction of Data and Processing Bandwidth

Data reduction is a necessary part of rapid image understanding systems as long as redundant information loss is associated with it. Data reduction is typically performed in feature based pattern recognition systems which can operate very fast but valuable information is often lost. The consequence of this information loss is to transform a potentially over-constrained vision problem into an under constrained one. Pixel based techniques are not lossy as are extracted features but they are very inefficient at exploiting global data dependencies. Processing cones or pyramids are an attempt to simplify the exploitation of global data dependencies in pixel oriented representations. Consequences of this approach include an increase in the amount of data that must be processed and the loss of semantically important detail in lower resolution processing levels.

In section IV.1 a new approach to image representation is described which features high fidelity data compression, reduced processing bandwidth for image understanding functions, and an ease in exploiting global context.

A control strategy found in animal vision systems for reducing processing bandwidth requirements is the indexing function or focus-of-attention operator. In any goal-oriented vision function, it should not be necessary to process each visual region with the same computational effort as every other visual region. As an example, empty sky may not receive the same attention from a working truck driver as the highway does even though the sky is within the driver's field of view. Indexing functions in the ESPI vision system achieve the reduced processing bandwidth requirement by sifting out only the most relevant and interesting parts of an image. Indexing functions are defined in greater detail in section IV.2

Principle of Modular Design

The existence of a modular organization in the human visual system is suggested by the apparent ease with which color blind or one-eyed individuals adapt to a colorful, 3-D world. If vision were not commonly an over-constrained problem and if human

vision were not modular, then one could not expect to make sense from very simple line drawings.

David Marr [Mar82] suggests that modularity found in biological vision systems was a prerequisite to the successful evolution of the human visual system. Since evolution depends on a series of incremental and successful mutations, each mutation must be localized in its effect. Otherwise, mutations successful in one area might have a devastating consequence on other areas of the organism. These theories may be very speculative, but that can not diminish the importance of modular design in creating complex vision systems.

In order to minimize process feedback, it would be helpful to partition the vision system into independent context domains. Feedback in image understanding systems is generally accepted to be a necessary and valuable feature. One example is the control of segmentation by higher level processes [Han78,Nag80,Naz84]. But process feedback carries with it a penalty. It either introduces a delay in system response or it causes synchronization errors. Synchronization errors result when feedback is computed from earlier imagery that differs from the imagery being affected.

IV. ESPI Vision System

The following three areas of active image understanding research are in various stages of development. The first one, a novel form of image representation, is relatively mature. The second area describes a control mechanism for distributed image understanding systems. The last area addresses the pragmatic problem of implementing complex AI software in distributed real-time hardware. Other active ESPI research areas that are not discussed here include 1) noniterative, reversible medial axis transformations, 2) constant-time search (with respect to the size of the database), 3) re-writing rules for multi-dimensional grammars, and 4) definition and application of (visual) scripts (after Shank) and routines (after Ullman,[Ull83]).

IV.1 Generalized Cylinder Picture Elements

The idea of the generalized cone or cylinder was introduced by Binford [Bin71] as a way of representing shapes in a computer. Other forms of shape description have been proposed also. A few of these include strip trees [Bal79], Fourier approximation of region edges [McKb..], pyramids or reduced resolution hierarchies, medial axis transformations [Ros66,Rut66],

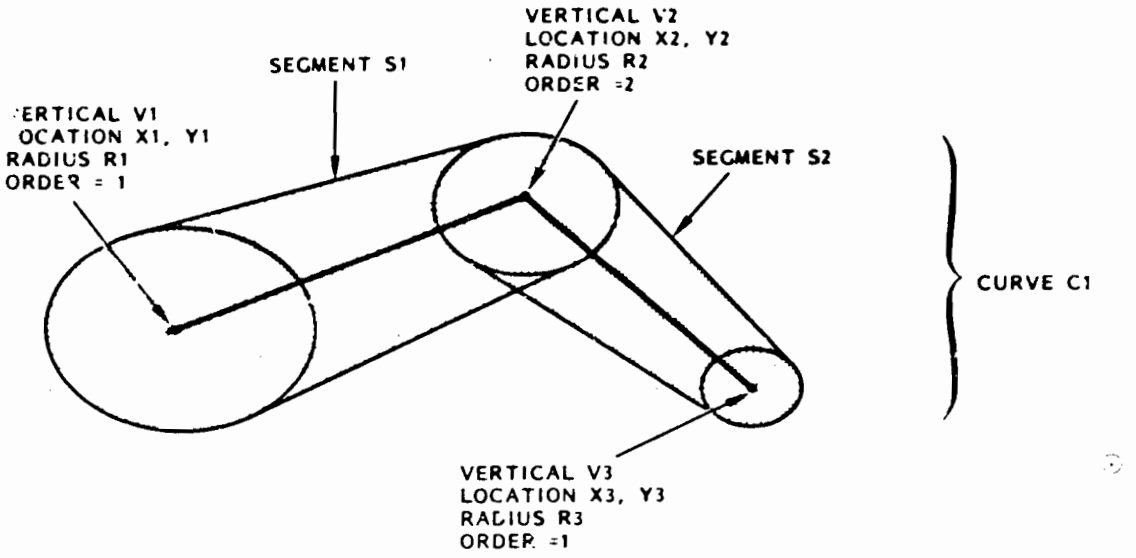


Figure 1. A Shape Represented by Two Gyxels

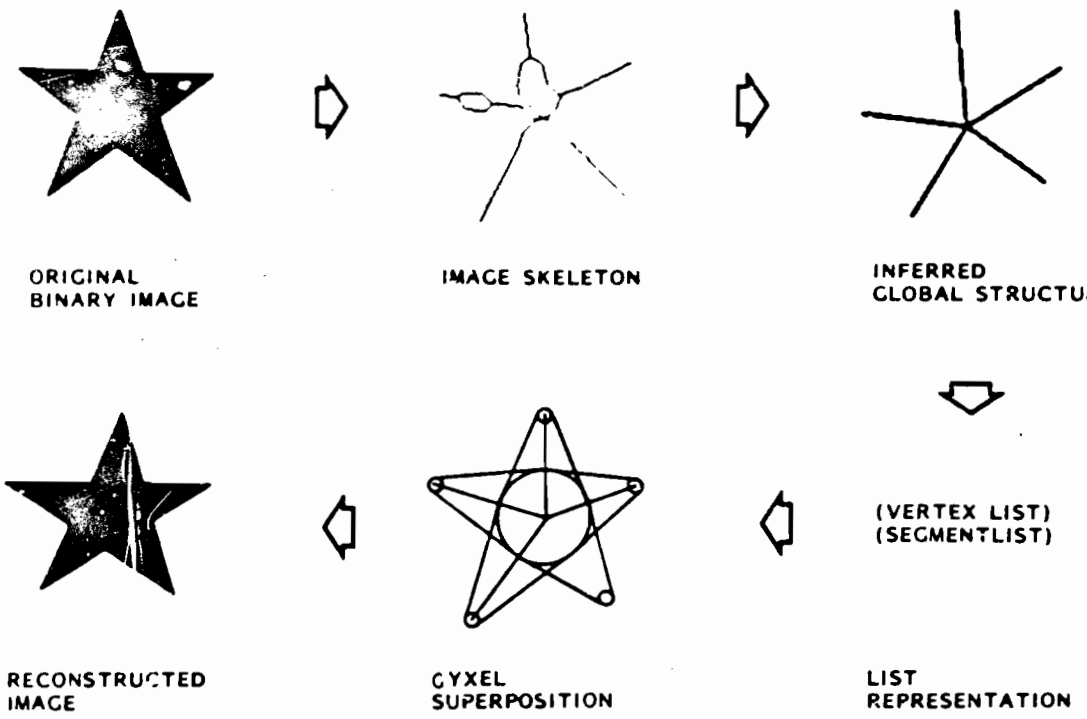
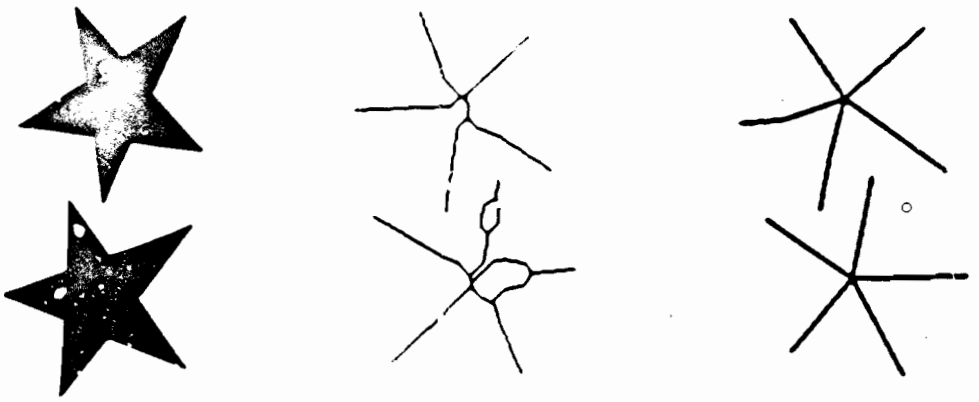


Figure 2. Pixel to Gyxe Transformation

and numerous quadtree representations. Each of these approaches attempt to rectify the problems that picture elements or pixels have in expressing global data dependencies.

We present a new form of image representation (not description) that combines features of the medial axis tranformation and the generalized cylinder. It is called the Generalized cYliner p(ic)ture ELement or gyxel. Each gyxel consists of a two-pulley/belt shape, shown in figure 1. Gyxels are constructed from binary images by performing a medial axis transform (MAT), parsing the skeleton into vertices and connecting segments, performing linear interpolations on the segments, removing small topological errors or artifacts of the MAT, then including interpolated width information into the vertex list. This is illustrated in figure 2. An image can be quickly recontructed from this list format to a frame buffer or the bit-mapped display of a workstation.

The gyxel provides a potentially lossless representation of greyscale images which are efficiently stored and manipulated in list notation. By adjusting linear interpolation error parameters (for curve and width interpolations) as well as the number of greyscale levels (or colors or textural types), one may trade off reconstruction fidelity for bandwidth reduction. An original 256x256 black-and-white image and its gyxel reconstruction are illustrated in figure 7. By performing segmentation labeling prior to transformation into gyxel format, individual gyxels may be labeled with respect to color, shade, or texture. LISP is the development language of choice for performing gyxel manipulation. In addition, gyxels, displayed on bit-mapped screens, may exist as objects with methods or processes attached to them. A more detailed treatment of gyxels and techniques for their manipulation is in preparation.



3a. 3b. 3c.

Figure 3. Correction of Structural Errors in Mat

Errors due to the linear MAT skeleton interpolation or the linear width interpolation can be adjusted if one wishes to trade off image fidelity for bandwidth. The image is represented in the form of a vertex list and a segment list. Each vertex element of the vertex list contains a vertex number, x and y locations, order (number of coincident segments), and radius. Each segment element of the segment list contains a curve number (shared by all segments of a connected set), a segment number, the originating vertex number, the destination vertex number, and a color/texture label. These lists are easily manipulated in the LISP programming language.

The medial axis transform has a major, well documented flaw which may explain why it has gained so little popularity. Similar shapes differing only by small edge perturbations or "holes" may account for very different skeletal shapes. One solution to this problem is to spatially low pass filter an image to the point where similar shapes will feature similar skeletons. This approach is not only information lossy, but often it does not work.

We have demonstrated a nonlossy approach which uses very simple, low level LISP functions to infer global structure from topologically different skeletons. Figure 3.a illustrates two star shapes: one of the star shapes contains two "holes". The resulting medial axis transformation of these two shapes is shown in figure 3.b. Note that even the "perfect" star does not result in a symmetric skeleton. By applying a series of very simple LISP functions to the segment and vertex lists representing the skeleton, it is possible to infer a five-spur symmetric structure. This is illustrated in figure 3.c. The reader is referred to [Rea85.2] for a complete description of this simple inference process. Although the example illustrates the reconstruction of a skeleton, a gyxel reconstruction would have been very simple.

2 Indexing Functions

Indexing [Ull83] is an operation used in vision systems to shift processing focus. By shifting processing focus to unique "odd-man-out" locations in an image, a vision system is able to formulate an initial hypothesis about the scene without

processing a potentially crippling amount of irrelevant data. In another sense, indexing functions select the most "interesting" data. In this way, they reduce the computational demands placed on hardware. Because the sifted data is so interesting, initial hypotheses are likely to be based on the most characteristic features of a scene. The likelihood of making mistakes may be less; this can result in less backchaining.

The existence of indexing functions in humans is suggested by psychological evidence and by physiological data. Indexing functions assume two very different forms: task-dependent and task-independent. Task-independent indexing has been called preattentive vision [Jul81]. It is characterized by being seemingly instantaneous, effortless, and is sensitive over a wide field of view. The perception of textural boundaries is only one example of preattentive vision. Attentive or task-dependent vision, on the other hand, is goal-directed. Goal-directed visual search is serial (much slower than preattentive vision) and usually is operative only over a relatively small field of view. Since the term "focus-of-attention" usually refers to attentive vision, we have selected "indexing" to refer to both attentive and pre-attentive forms.

Although several image understanding systems exist which use focus-of-attention operators [Bro82,Naz84], they are goal-oriented or task-dependent. The ESPI vision system combines task-dependent and task-independent focus-of-attention operators. This provides multi-level bandwidth reduction for low (task independent) level processes as well as higher (task dependent) processes.

One example of preattentive vision that will be integrated into the ESPI vision system is the YATA texture discrimination algorithm [Rea85.1]. The YATA algorithm models Julesz' Texton theory of preattentive vision [Jul81]. It is able to discriminate two different textures having identical Fourier power spectra (and so, identical intensity mean and variance distributions). This is demonstrated in figure 4. Detection of subjective contours by the YATA algorithm is illustrated in figure 5.

An early example of two goal-directed indexing functions is shown in figure 6. The unordered segment list and vertex list are filtered with respect to qualities which might suggest "obvious" roads or buildings. The objects identified in figures 6.b and 6.c are hypotheses which may then be tested in a distributed multi-processing environment (note: this example uses an early reconstruction form using constant width gyxels). Once these hypothetical objects are validated, they are used to suggest the location of less obvious roads and buildings. This incremental approach serves two purposes: a data reduction decreases computational bandwidth and there is less chance of mistake or backchaining, because the process begins with hypotheses which are least risk or most obvious (like road shapes found in aerial vision applications).

IV.3 Distributed Artificial Intelligence

It is our belief that the mapping of complex AI-type functions onto general purpose multiprocessing systems for real-time applications is not only risky but unnecessary. The overhead due to interprocessor communication is likely to be much greater in symbolic processors than it has been in numerical processors. Two recent developments have made the prospect of custom hardware practical and cost effective. The DARPA sponsored MOSIS circuit fabrication process will soon be able to turn CAD designs into tested silicon within 30 days. Object oriented languages or functional languages provide a software development and simulation environment. Moreover, once a program written in an object oriented language is debugged, it may be mapped directly into virtual machine/FIRMWARE CAD decriptions. In several years, it should be possible to go from a software simulation to working silicon in much less than one month. It will be a cheaper, faster, and more reliable approach to prototyping than the current approach. Since the architecture is customized, there will be much less interprocess communication overhead than would be found in general purpose machines.

The ESPI vision system is being developed in an object oriented language. This language is currently being prototyped in LISP. The Smalltalk-80 programming language [Gol83, Gol84] has served as a model for the implementation of this simulation/development language. The definition of the virtual machines and implementation feasibility studies remains to be done. The virtual machines will probably include RISC computers as well as simpler automata. The simulation (as well as the implementation) will reflect many of the design issues promoted for data flow machines [Den81] and parallel architectures for AI [Hew84].

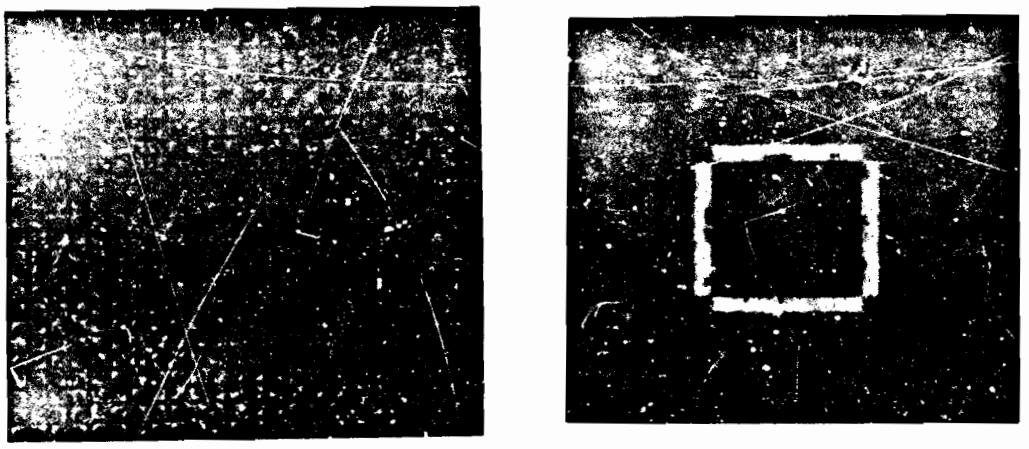


Figure 4. Discrimination of Textures Having Identical Fourier Power Spectra

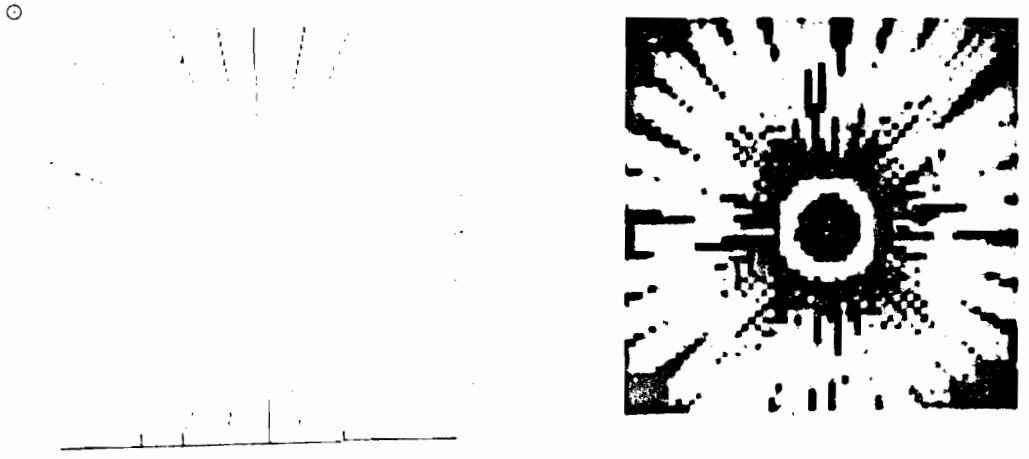
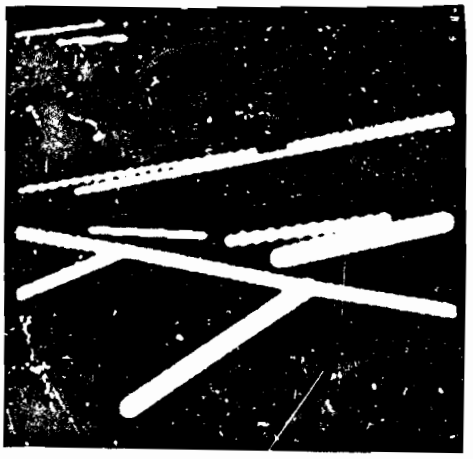


Figure 5. Detection of Subjective Contours

a. ORIGINAL IMAGE

b. BUILDING-LIKE OBJECTS



c. ROAD-LIKE OBJECTS

Figure 6. Example of Simple Indexing Functions

V. Conclusions

In evaluating the feasibility of real-time image understanding systems, the ESPI vision system is testing a variety of novel approaches to shape and knowledge representation, image understanding control mechanisms, and implementation schemes. The potential for additional technological breakthroughs seems promising.

Significant breakthroughs have been made to date. These include the first rigorous computer inplementation of the Texton Theory of Preattentive Vision. This algorithm demonstrated the successful detection of subjective contours and the discrimination of two different textures having identical Fourier power spectra [Rea85.1]. A syntactical method for removing topological errors common to the medial axis tranformation was also demonstrated [Rea85.2]. Finally, this paper introduces a lossless representation of binary images which is efficiently stored and manipulated in list structures.

Figure 7. 256 x 256 Pixel Image and Gyxel Reconstruction

References

[Bal79] D.H. Ballard, "Strip trees: A Hierarchical Representation for Map Features," PROC. IMAGE UNDERSTANDING WORKSHOP, April 1979, pp.121-133.

[Bin71] T.O. Binford, "Visual Perception by Computer", presented at the IEEE Syst., Sci., Cybern. Conf., Miami, FL, invited paper, Dec. 1971.

[Bro79] R.A.Brooks, R. Greiner, and T.O. Binford , "The ACRONYM Model-based Vision System", PROC. IJCAI-6, Tokyo, Japan, August 1979, pp. 105-113.

[Bro82] Roger A. Browse, "Knowledge-based visual interpretation using declarative schemata," PhD. thesis/ Tech. Rep. TN 82-12, Univ. British Columbia, Dept. Comp. Sci., Nov. 1982.

[Clo71] M.B. Clowes, "On Seeing Things", ARTIFICIAL INTELLIGENCE, Vol. 2, Issue 1, 1971, pp.79-116.

[Den84] Jack B. Dennis, "Data Should Not Change: A Model for a Computer System", MIT Laboratory for Computer Science, Computation Structures Group Memo No. 209, July 1981.

[Gol83] Adele Goldberg and Daniel Ingalls, SMALLTALK-80: THE LANGUAGE AND ITS IMPLEMENTATION, Addison-Wesley, Reading, MA, 1983.

[Gol84] Adele Goldberg, SMALLTALK-80: THE INTERACTIVE PROGRAMMING ENVIRONMENT, Addison-Wesley, Reading, MA, 1984.

[Guz68] A. Guzman,"Decomposition of a Visual Scene into Three- Dimensional Bodies", AFIPS PROCEEDINGS FALL JOINT COMP. CONF., Vol.33, 1968.

[Hal85] R.P. Hall and D.F. Kibler, "Differing Methodological Perspectives", AI MAGAZINE, Vol. 6, No. 3, Fall 1985, pp. 166-178.

[Han78] A.R. Hanson and E.M. Riseman, "VISIONS: A Computer System for Interpreting Scenes", COMPUTER VISION SYSTEMS, Academic Press, A.R. Hanson and E.M. Riseman, (Eds.), New York, 1978, pp. 303-333.

[Hav80] W.S. Havens and A.K. Mackworth, "Schemata-based Understanding of Hand-Drawn Sketch Maps", PROC. THIRD CONF. CANADIAN SOC. COMP. STUDIES INTEL., Victoria, Canada, 1980, pp. 172-178.

[Hew84] Carl Hewitt and Henry Lieberman, "Design Issues in Parallel Architectures for Artificial Intelligence", MIT AI Laboratory, Memo No. 750, November 1983.

[Mar82] David Marr, VISION: A COMPUTATIONAL INVESTIGATION INTO THE REPRESENTATION AND PROCESSING OF VISUAL INFORMATION, W.H. Freeman, San Francisco, 1982.

[McK85] D.M. McKeown, Jr., "Alignment and Connection of Fragmerted Linear Features in Aerial Imagery", PROC. IEEE COMP. SOC. COMP. VISION PAT. RECOG., San Francisco, June 1985, pp. 55-61.

[Nag80] M. Nagao and T. Matsuyama, A STRUCTURAL ANALYSIS OF COMPLEX AERIAL PHOTOGRAPHS, Plenum Press, New York, 1980.

[Naz84] A.M. Nazif and M.D. Levine, "Low Level Image Segmentation: An Expert System", IEEE TRANS. PAMI, Vol. PAMI-6, No. 5, Sept. 1984, pp. 555-577.

[Rea85.1] T.C. Rearick, "A Texture Analysis Algorithm Inspired by a Theory of Preattentive Vision", IEEE COMP. SOC. PROC. COMP. VISION PATTERN RECOG. '85, San Francisco, CA, June 1985, pp. 312-317.

[Rea85.2] T.C. Rearick, "Syntactical Methods for Improvement of the Medial Axis Transformation", PROC. SPIF: APPLICATIONS OF AI II, J.F. Gilmore, ed., Arlington, VA, Vol. 548, (1985), pp.110-115.

[Rob65] L.G. Roberts, "Machine Perception of Three-dimensional Solids", OPTICAL AND ELECTRO-OPTICAL INFORMATION PROCESSING, J.T. Tippett et al. (Eds.), MIT Press, Cambridge, Mass., 1965, pp.159-197.

[Ros66] A. Rosenfeld and J.L. Pfaltz, "Sequential Operations in Digital Picture Processing", JOUR. ACM, Vol. 13 (1966), pp. 471-494.

[Rut66] D. Rutovitz, "Pattern Recogntion", JOUR. ROYAL STATIS. SOC., Vol. 129(1966), pp.504-530.

[Sha72] R.C. Shank, "Conceptual Dependency: A Theory of Natural Language Understanding", COGNITIVE PSYCHOLOGY, Vol. 3, No. 4 (1972), pp.552-631.

[Sha73] R.C. Shank, N. Goldman, C. Rieger, and C. Riesbeck, "MARGIE: Memory, Analysis, Response Generation and Inference in English", PROC. 3D IJCAI, 1973, pp. 255-261.

[Ull83] Shimon Ullman, "Visual Routines", MIT AI Laboratory, Memo No. 723, June 1983.

GC970145-12

DYNAMIC ARCHIVAL SCENE MODEL

Hatem N. Nasr
Raj K. Aggarwal
Durga P. Panda

Honeywell Inc., Systems and Research Center
Minneapolis, MN 55440

ABSTRACT

This paper describes a blackboard architecture for representing knowledge-base and scene information for outdoor multiscenario dynamic scene interpretation. The blackboard, called Dynamic Archival Scene Model (DASM), allows intelligent compression of sensed image information into an efficiently structured fashion. Scene context information is also incorporated into the blackboard. The DASM provides easy access to the scene model from all the cooperating processing levels of the vision system performing scene interpretation. The cooperating functional modules can either dynamically update the information in DASM or acquire information from DASM to perform their individual functions. The end result is an invariant representation of the scene dynamics for knowledge based interpretation of outdoor temporal events.

I. INTRODUCTION

Interpretation of outdoor dynamic scene under variable scenario conditions and diverse atmospheric ambients is difficult because dynamic object and relational representations are difficult to define, and low-level vision algorithms often provide ambiguous or incomplete primitives from the sensed image (Rosenfeld 1984). This makes dynamic scene interpretation highly vulnerable to variations in the image content and quality.

There have been efforts to build world models that provide invariant domain knowledge for vision systems (Mackworth 1981). Brooks et al (1979) in their top-down prediction-hypothesis-verification paradigm emphasize object driven modeling rather than data driven modeling. Integrated top-down and bottom-up ap- proaches (Matsumaya 1985) begin to ad- dress the need for mapping of the dynamic image representation to a world model. Feldman et al (1978) and Nagoa et al (1980) use a novel approach to knowledge directed image analysis in aerial imagery. Their model offers encouraging results, however, their image models assume a somewhat static environment and ideal quality imagery. Hanson et al (1980) discuss image models as an instantiation to a symbolic world model in what they refer to as short-term and long-term memory. The model has many interesting capabilities. However, the results are sensitive to variations in the image quality and content. Further detailed survey of models for vision systems can be found in (Binford 1982).

Reliable and robust interpretation of outdoor scene requires an approach to integrating the static domain knowledge, with models of temporally archived distortion invariant information, extracted from image data (Fischler 1973). We discuss here a model, called Dynamic Archival Scene Model (DASM) with emphasis on efficient image data representation of dynamic and archival information. Updating, information recovery, and integration of the dynamic information with the world knowledge are also encompassed functions in the DASM.

DASM provides a highly structured and abstracted image model, built on a blackboard architecture framework.

Some key features of this model are:

1. Symbolic image descriptors coupled with contextual information compensate for incomplete world model representations and weak low level vision (e.g. segmentation and edge detection) processes. Loss of details in the data is accounted for in the integration of the information.
2. DASM creates a centralized image data-and-knowledge-base that serves as an interface for all processing levels (low, intermediate, and high). This interface facilitates top-down feedback control of vision processes.
3. It provides a compatible representation of the dynamics with the static world model. This reduces the search space for semantic matching of relationships in the dynamic scene with the expected models.

In the next section, we first preview knowledge representation in the static world model. In the following section, we present an overview of DASM.

II. THE STATIC SCENE MODEL

The Static Scene Model (SSM) contains static knowledge describing expected scene objects and regions. It is a schema representation of a semantic network implemented as a frame-based structure where slots, in the frames, correspond to attributes describing world objects and relations.

The SSM describes regions and objects expected to appear in a specified world domain. For example, roads, vehicles, bridges, vegetation, trees, sky. Many similar representational approaches have been taken in the IU community, only the particular domains vary.

Regions and objects included in the Static Scene Model share a certain set of common attributes that describe them under general conditions. These attributes are described in contextual and relational manner. Shape, relative size, relative position and texture are examples of such attributes (or features). These attributes are common to all objects. Figure 1 illustrates the content of SSM. In the Static Scene Model the attributes have expected values that are constrained by the particular operating scenario, such as ground-based sensor, airborne sensor, etc.

These models in the Static Scene Model are generic objects and regions schemata, where the attribute values in the models are a superset of the ones in the images. In other words, the image model contained in the DASM is an instance model of the SSM.

Knowledge represented in SSM corresponds to the intrinsic, contextual and relational characteristics of objects and regions. These characteristics are explored by identifying similarities differences, and uniqueness of these objects under similar and different scenarios.

Given a particular scenario, every region or object has a set of geometrical, physical, contextual and relational properties that is always valid, under minor variations. These different properties are described as follows:

- Geometry: Shape (straight line segments, quadrilateral, circular), concavity and convexity, three-dimensional to two dimensional signature, perimeter, rotation and translation.
- Physical properties: Reflectivity, absorbitivity, intensity, material composition, and spectral characteristics.
- Contextual and relational properties: Expected image location, size, neighboring regions, inside-of other regions, and containing other regions or objects.

Contextual information introduced in the SSM directly relates to the image domain. However, some information is not derived from the image

data. This includes ancillary information such as weather condition, goal-driven object search, and others. For example, not all regions are produced by the same segmentation algorithm. The same image is passed through different segmentors, which are specialists in extracting convex regions, elongated regions, etc. This specific region segmentor information is part of the expected contextual information incorporated in each object and regional schema in the Static Scene Model.

III. THE DYNAMIC ARCHIVAL SCENE MODEL

The DASM consists of spatial and temporal information, low level primitives, as well as symbolic and contextual information. The DASM also contains a dynamically updated historical scene buffer. Figure 2 shows the system level block diagram. Regional scene representation in DASM is illustrated in Figure 3.

Local Schema

The DASM representation is based on a collection of schemata and is implemented in a frame-based data structure. It contains attributes (called slots), similar to SSM, and their corresponding values which describe image region features and their relationship to other image regions as shown in Figure 4.

There are two levels of image information abstraction as illustrated in Figure 2. The first level is a reorganization of low-level feature vector data into a frame-based data structure, with no assumed inheritance. The second level contains symbolic descriptors, extracted from the first level, represented in a frame-based structure. The first level serves as an input to the historical scene buffer where primal data is used for compiling temporal and spatial information.

Every schema in DASM corresponds to a specific image region. The collection of these schemata creates an instance model of the Static Scene Model. Every slot in each frame corresponds to an image feature such as: area, shape measure, centroid, location, length-to-width ratio, texture measure, etc. Some of the common region attributes are:

TEXTURE = SMOOTH, COARSE, ETC.
SHAPE = ELONGATED, TRIANGULAR, SQUARE, CIRCULAR, IRREGULAR
RELATIVE LOCATION = BEHIND, IN-FRONT, NEXT-TO, ON-TOP-OF, UNDER
ABSOLUTE LOCATION = FAR-AWAY, MID-RANGE, NEAR, FOREGROUND, BACKGROUND
RELATIVE SIZE = LARGE, MEDIUM, SMALL
ABSOLUTE SIZE = LARGE, MEDIUM, SMALL

Some of the above attributes are the result of associating and interpreting two or more numeric features. For example, the AREA and RANGE produce ABSOLUTE-SIZE, as shown in Figure 5.

Primal Model

The primal model, called Archival Scene Model (ASM) is a central database that contains all the relevant primal information about the current scene, such as intermediate image results, range and transforms estimates as well as information pertaining to individual regions such as: features, silhouettes, and tracking data. Multi-frame information at the image, region, and segment levels is also stored in the ASM.

Historic Buffer

DASM contains archived information from a temporal sequence of image frames which form the historic scene buffer. The data is organized under a scene number and corresponding regions. Figure 6 illustrates how such a buffer is constructed. This historical scene buffer serves as a reference for object motion detection, prediction, and tracking. It also provides a mean for compiling temporal and spatial information.

DASM as an Interface for Feedback

An interface is a channel of communication between different levels of processing, a basic requirement for feedback control. DASM establishes that interface as shown in Figure 7.

The interface supplies information and data for feedback between different levels of processing. The data required for the interface is determined by analyzing the inputs and outputs of the feedback. The feedback from mid to low level processing consists of labeled regions and objects classifications. Outputs from low level processing are algorithms and parameter selection. The interface is regional scene data from the DASM, ancillary information and confidence measures of labeling. This information will feed into feedback production-rules which transform it into low-level commands. An example would be:

- IF ROAD is found THEN SEARCH for VEHICLES
- IF GOAL = SEARCH for VEHICLES THEN SEARCH for CONVEX objects
- IF SEARCH for CONVEX objects THEN run BACORE (segmentation) at WINDOW_X

The road found in above example, schema is accessed from the DASM and input is provided to calculate the window size.

IV. EXPERIMENTS WITH DASM

The Dynamic Archival Scene Model, along with the Static Scene Model, is an integral element of knowledge based vision system for outdoor scene interpretation. The model interface includes the discrete vision functional modules, such as synthetic stereopsis, optical (segment free) motion detection, symbolic motion detection, multi-object tracking, and object identification. The interface also includes low and intermediate level vision operators such as edge operators, line operators, texture segmentors, and scene transform operators. Lastly, the interface also includes the knowledge-based controller of the vision system. Thus the DASM acts as an efficient central link among the various elements of the vision system.

Incorporation of the model into the vision system helps to resolve conflict and ambiguity in the interpretation of scene elements (such as object identity) over successive temporal image frames. It semantically confirms the scene dynamics interpretation (such as object motion) by archiving dynamics information and evaluating it with temporal and spatial context. Experimental results show consistent and robust output of image frame sequence analysis in the form reliable object identification, minimal object misclassification, consistent motion detection and correspondence, and temporal pattern detection. Figure 8 illustrates the results over a time sequence of image frames, where DASM information helped in robustly detecting and identifying objects of interest, estimating the dynamic sensor and object motion parameter, and tracking the objects in the scene. The vectors overlayed over the objects in Figure 8 indicate the predicted direction and motion of objects over time.

V. CONCLUSION

We presented an approach to integrated modeling of the Static Scene Model and dynamic image abstraction model in the form of Dynamic Archival Scene Model. The integrated model helps facilitate correspondence of image abstracted information with expected world knowledge. Experimental results show reliable interpretation of dynamic scene in the presence of ambiguous and incomplete low level operator output and in presence of both spatial and temporal noise.

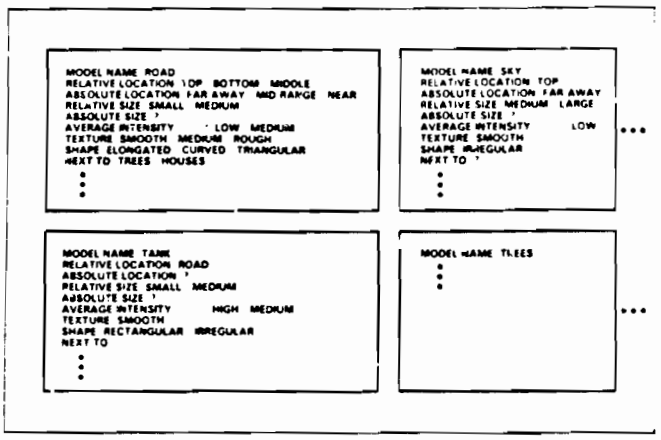


Figure 1. Representation in the Static Scene Model

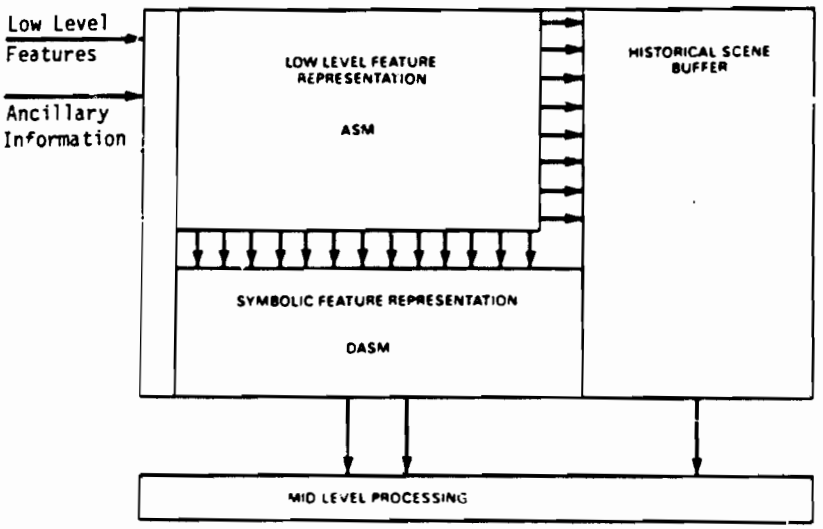


Figure 2. DASM System Level Block Diagram

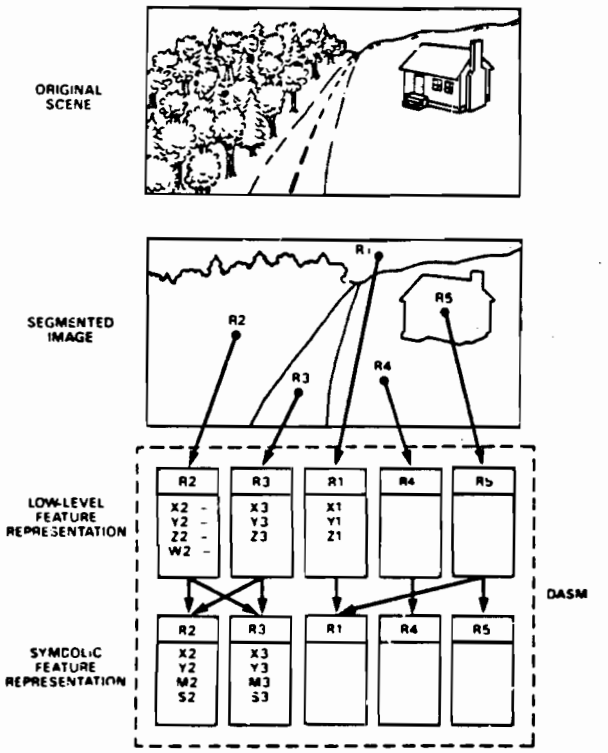


Figure 3. Region Representation in DASM

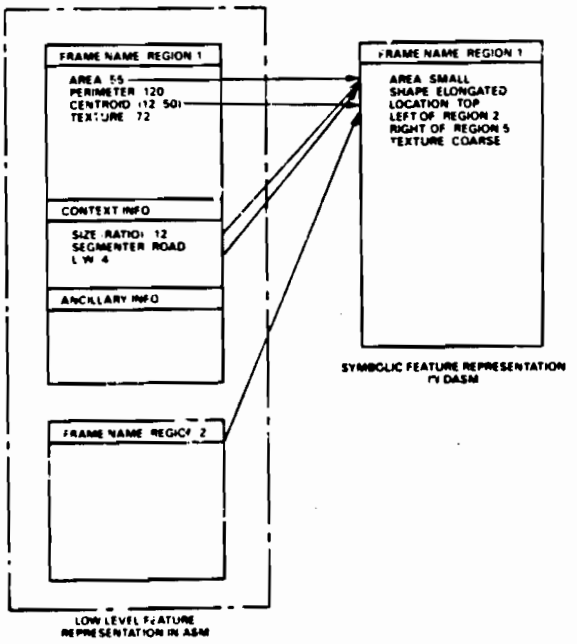


Figure 5. Sources of Symbolic Features

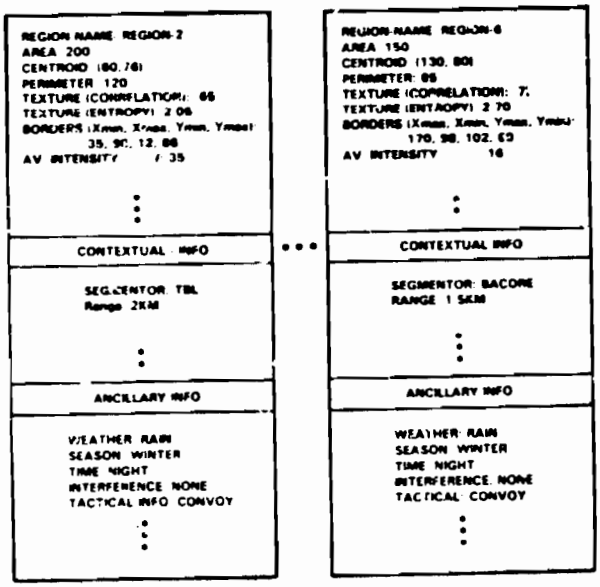


Figure 4. Low-Level Feature Representation in DASM

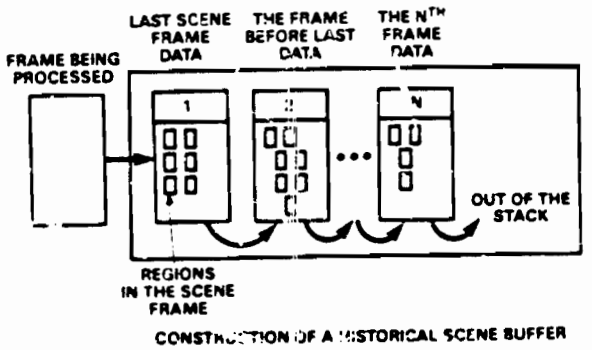


Figure 6. Historical Scene Buffer in DASM

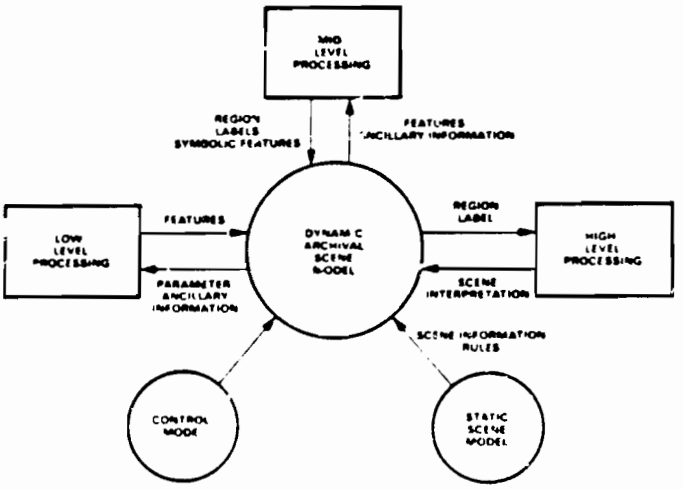


Figure 7. DASM: Interface Between Low Level, Mid Level, and High Level Processes

[illegible] for a variety of images

REFERENCES

T.O. Binford (1982), Survey of Model-Based Image Analysis Systems, The International Journal of Robotics Research, Vol. 1, No. 1, pp. 18-64.

R. Brooks, R. Cereiner, and T. Binford (1979), The ACRONYM Model-Based Vision System, IJCAI Proceedings, Tokyo, p. 105.

J.A. Feldman, D.H. Ballard, and C.M. Brown (1978), An Approach to Knowledge-Directed Image Analysis, in Computer Vision Systems, ed. A. Hanson and E. Riseman, New York: Academic.

M.A. Fischler and R.A. Eschlager (1973), The Representation and Matching of Pictorial Patterns, IEEE Trans. on Computers, C-22, January.

A.M. Hanson, C.C. Parma, and E.M. Riesman (1980), Experiments in Schema-Driven Interpretation of a Natural Scene, Coins Tech. Rept. 80-10, Amherst, Mass.: Univ. of Massachusetts.

A.K. Mackworth and W.S. Havens (1981), Structuring Domain Knowledge for Visual Perception, IJCAI Proceedings, Vancouver, p. 625.

T. Matsumaya and V. Hwang (1985), SIGMA: A Framework for Image Understanding, IJCAI Proceedings, California, p. 908-915.

M.L. Minsky (1975), "A Framework for Representing Knowledge", in the Psychology of Computer Vision, ed. by P.H. Winston, McGraw-Hill.

M. Nagoa and T. Matsuyama (1980), A Structural Analysis of Complex Aerial Imagery, Plenum.

A. Rosenfeld (1984), Image Analysis: Problems, Progress and Prospects, Pattern Recognition, Vol. 17, No. 1, pp. 3-12.

CC9706145-13

IMAGE UNDERSTANDING RESEARCH AT GENERAL ELECTRIC

J. L. Mundy[1]

General Electric
Corporate Research and Development
Schenectady, New York

INTRODUCTION

The current research at General Electric in image understanding has evolved out of a long period of experience in building systems for automatic visual inspection in a factory environment [Mundy and Porter 1983]. Out of this experience has come the opinion that there are a number of limitations in image understanding technology that prevent its application to general environments.

The primary weakness in the technology is that many assumptions have to be made about the geometric and signal processing constraints associated with a given problem in image analysis or object recognition. The basic algorithms are fragile in the sense that if the environment does not agree with the assumptions within a small margin, then the results are unpredictable and usually unsatisfactory. On the other hand, if the constraints can be satisfied, then it is possible to perform rather sophisticated recognition tasks with excellent reliability [Mundy and Joynson 1977]. A system developed to inspect small manufactured parts was able to achieve an error rate of less than 0.1% as determined over several years of operation.

In 1983, a new program was initiated to study fundamental issues in image understanding. The goal of the program is to explore new techniques that show promise in extending robustness and flexibility of image understanding systems. This program has two main technical thrusts. The first is aimed at evolving new techniques in model-based vision. The second area of interest is geometric reasoning and its application to image understanding. This program emphasis is based on the opinion that these are the most important research issues for rapid progress in image understanding. The paper will outline our ideas for a model-based image understanding system as well as the application of geometric reasoning to such a system.

MODEL-BASED VISION

The use of geometric models to locate and recognize objects in scenes has proved surprisingly effective in dealing with cluttered scenes with unreliable feature extraction [Brooks 1981], [Goad 1982], [Ayache 1983], [Grimson and Lozono-Perez 1985]. The constraints provided by the model are a very effective filter that can eliminate many incorrect interpretations. The match does not have to be complete since the probability of more than a few features matching is quite low, unless the assignment is actually correct.

The concept of model-based vision and the role of geometric reasoning are illustrated in Figure 1. This system block diagram indicates the major functions in a system that uses geometric models to identify and locate objects. First, we describe the basic aspects of this model-based vision system that will guide our development over the next few years.

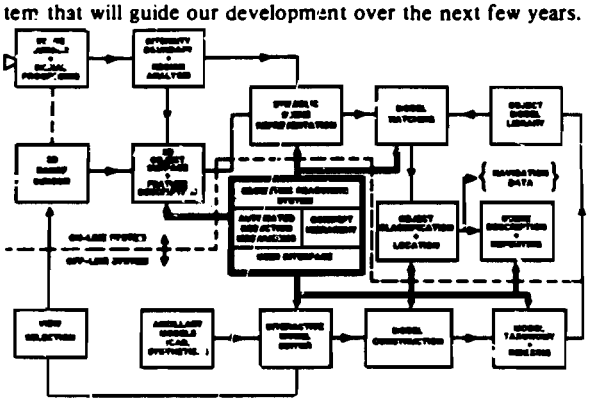


Figure 1. The overall system block diagram. This diagram indicates the important functions in a model-based image understanding system.

The system is divided into two major sections, an on-line system and an off-line system. The on-line partition has the basic function of recognizing objects by matching a relational model of the object with a symbolic representation of the scene. In the proposed effort, both the scene description and model will be primarily geometric descriptions with a small set of ancillary properties such as surface texture and reflectance. The off-line portion is responsible for the formation of models and model libraries as well as the development of efficient matching rules and strategies that are needed to support the on-line activities.

The current hardware configuration is shown in Figure 2. We are using the Symbolics 3600 Lisp Machine with frame grabber and color display. We have built our image-understanding software on top of IMAGE-CALC which is an excellent development environment for image processing developed by Lynn Quam at SRI [Quam 1984].

1. The work reported here has involved cooperative efforts within the Logic and Inference Systems Program at General Electric Corporate Research and Development. Contributions to this work have been made by M. Barry, C. Connolly, R. Jaenicke, D. Kapur, H. Ko, D. Musser, P. Narendran, R. Stenstrom, R. St. Peters, and D. Thompson.

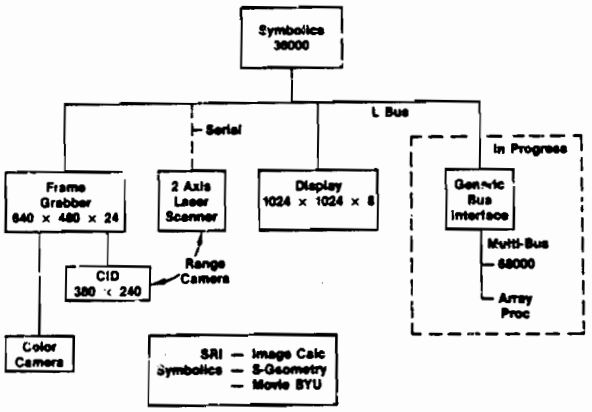


Figure 2. The hardware configuration for the image understanding system.

SYMBOLIC SCENE REPRESENTATION

The symbolic scene representation is obtained from either a visual image sensor or a 3-D range sensor. A low level boundary description is obtained from each sensor and segmented into a 3-D solid surface representation. This process involves standard algorithms for edge detection and linear boundary segmentation. In the case of multiple intensity views, stereo matching is used to extract a direct 3-D description. This information is augmented by direct range data obtained from the range sensor. It is planned to experiment with various combinations of sensor information, including the standard case of a single intensity image. The final symbolic scene representation consists of line segments that are interpreted as intensity or range boundaries and regions, which are connected sets of uniform texture or surface elements.

MODEL MATCHING

The second main process in the on-line system is the retrieval and matching of an object model with the symbolic scene description just described. In this system, the matching process is considered to be mainly an identification of the perspective coordinate transformation that maps a three-dimensional solid surface model into a 2-D scene description. The determination of the transformation is carried out by matching elements of the scene with elements of the model. In the simplest case, these elements would be line segments that are matched according to orientation and position.

2-D Model Matching

We have completed the development of a 2-D model matching system which acquires models from the 2-D scene itself. The techniques we have employed are quite similar to other 2-D matching systems [Grimson and Lozono-Perez 1984]. One distinguishing aspect of our experiments has been the use of color information.

The matcher is based on straight line segments that are grown from intensity edges found by the Canny edge detector [Canny 1983]. The line segments are determined by a combination of curvature extrema [Asada and Brady 1984] and a deviation tolerance [Tomek 1974].

The color information is introduced after the object boundary segments have been determined. The color image is transformed into intensity, hue, and saturation components. The image segmentation is based on intensity; however, both sides of the line segment are labeled according to the mean and standard deviation of hue and saturation.

In this manner the "side" of a line can be matched to a "side" in the model with a similar color. Those regions with low saturation are not filtered by color, since hue is not a reliable feature in that case. Also, if a hue data has a high standard deviation, the hue value is not considered to be very significant.

It is not straightforward to determine a distance metric to measure the agreement of the hue and geometric properties of a line segment. In the current program, the line segments are ranked by sorting the segments separately on each measurement value (in the current case length and hue). The sorted list is scanned until scene segments are found that are as close as possible to a given model segment within a given tolerance band. This ranking process serves to eliminate irrelevant matches of the model segments into the scene.

The proposed assignments from the scene into the model are used to define clusters in the space of model-to-scene transformations [Ballard 1981]. The transformation space for 2-D matching has three dimensions, corresponding to one rotational and two translational degrees of freedom. In the current algorithm, rotation is handled separately from translation. The cluster in transformation space, which is the most compact and contains the most matches, is used to determine a mean transformation vector.

The match determined by this mean transformation is checked for satisfactory agreement between the scene segments and the model segment. In this case, agreement is based on the number of matched segments and the normal distance error between the model and the scene segment positions. If the agreement is not satisfactory, the next most attractive segment is selected and tested.

It is emphasized that the association of model segments and scene segments is not implemented as a tree search. A group of features are associated with each model segment and are then clustered in transformation space. Earlier experiments with a tree search algorithm showed that it is difficult to specify a sequential priority ordering on the quality of feature matches.

A sample of 2-D matching is illustrated in a series of figures. Figure 3 shows an intensity image of a typical scene used in these experiments. The final result of processing the scene into line segments is given in Figure 4 and is superimposed on the intensity data in Figure 5. The assignment of model-to-scene segments at an intermediate stage of the match is shown in Figure 6. The final match is shown in Figure 7. This matching technique is able to find objects in cluttered scenes where many of the object features are missing or fragmented.

3-D Matching

A new experiment is currently under way to evolve to three-dimensional polyhedral models and to match into two-dimensional perspective intensity or color images. At this time, the use of both line segments and vertices as features is being explored. At present the 3-D models are created by CAD techniques. Figure 8 shows the assignment of model vertices into 2-D scene vertices. The scene vertices are obtained by extending scene segments until they intersect.

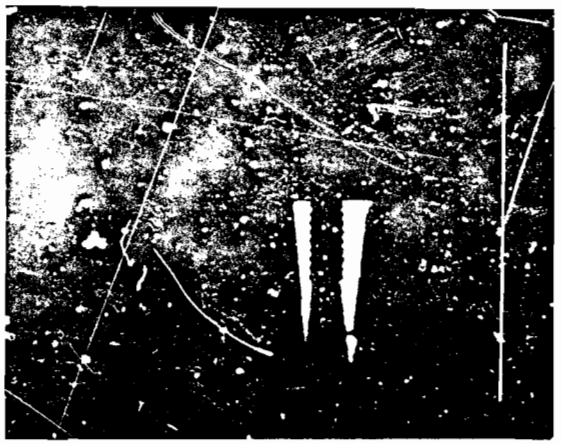


Figure 3. Image intensity for a typical scene used in the 2-D matching experiments.

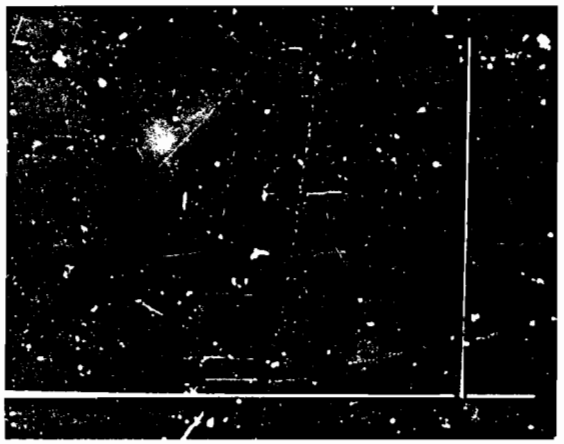


Figure 4. The edge segments produced by applying the Canny edge detector and a linear segmentation algorithm to Figure 3.

Figure 5. The line segments superimposed on the intensity data of Figure 3.

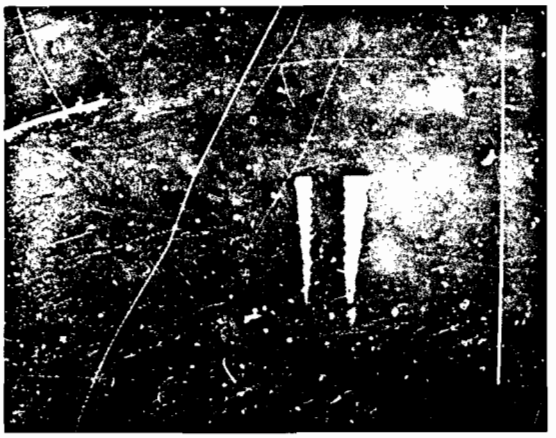


Figure 6 An intermediate match between the model and the scene.

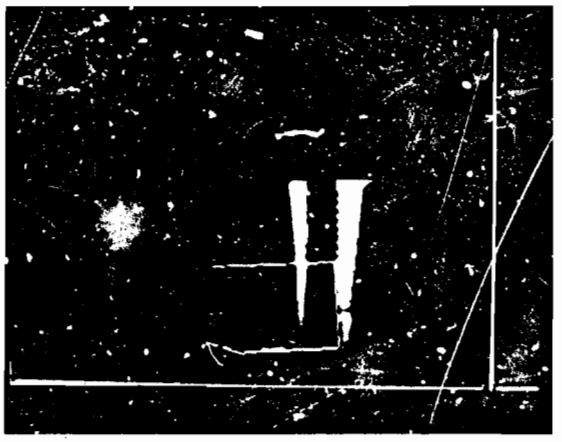


Figure 7. The final match position. The model is now within tolerance limits to the line positions in the scene.

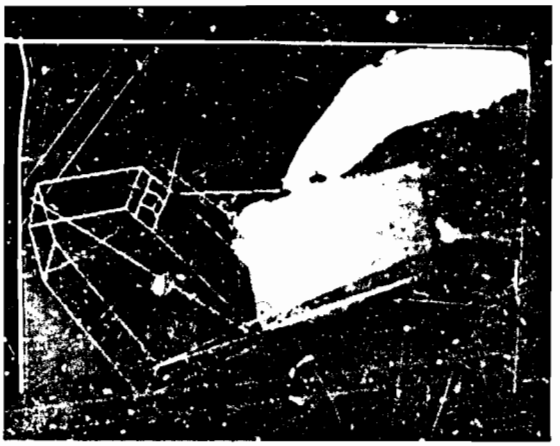


Figure 8. A 3-D matching experiment based on vertex assignments. The match is shown in an intermediate stage of proposed assignments.

OBJECT MODEL LIBRARY

It is expected that a relatively large number of models will be required to analyze the types of environments encountered in the autonomous vehicle application. For example, road obstacles could arise from a large variety of objects both natural and man-made. The need for a library of models raises the question of how to organize and index the models so that the model matching phase can be carried out efficiently.

The development of a model taxonomy and indexing scheme is another interesting application of the technology we are developing for geometric reasoning. We propose that a hierarchy of geometric concepts serves as a useful starting point to develop model characteristics that are effective indices into the library. For example, the distinction between convex and non-convex can be determined from an analysis of the scene on the basis of range-jump boundaries and occlusion edges.

Other geometric properties such as colinearity, symmetry, and genus should also be useful in classifying the models. The invariance of these properties to perspective transformation must also be determined in order to evaluate their effectiveness for model indexing.

OBJECT DESCRIPTION AND REPORTING

After the successful match of a model into the scene it is possible to answer queries about the scene and to report information needed for navigation and planning. It is expected that the sensed data provides at best a fragmentary description that must be augmented by information derived from the model. The simplest example is the use of 3-D information from the model to describe portions of the object that are not visible. Such information is useful in planning vehicle paths around road obstacles.

The matching of landmarks in the scene allows reference to maps and terrain data, which is useful for navigation. It is likely that the matching process here is much easier than for obstacles and tactical vehicles. The process is characterized by tracking a known object under a small range of possible transformations. Once the landmark has been located, a simple predictive model for vehicle motion can update the perspective transformation.

THE MODEL FORMATION PROCESS

The main activities to be carried out in the off-line portion of the system are model construction and geometric reasoning. The goal of these processes is to create models and matching rules to support efficient on-line object recognition.

There are two main approaches to the formation of models under investigation at present. The first involves a manual process where a wire-frame or surface model is created using a CAD solid modeling system. This model is based on a priori information such as a set of mechanical drawings for the object, or is partially based on measurements taken from images or from the object itself. In our current implementation, we are using the Symbolics S-Geometry package as a means for creating 3-D models. An example of model editing is shown in Figure 9.

The second approach is based on direct learning from 2-D

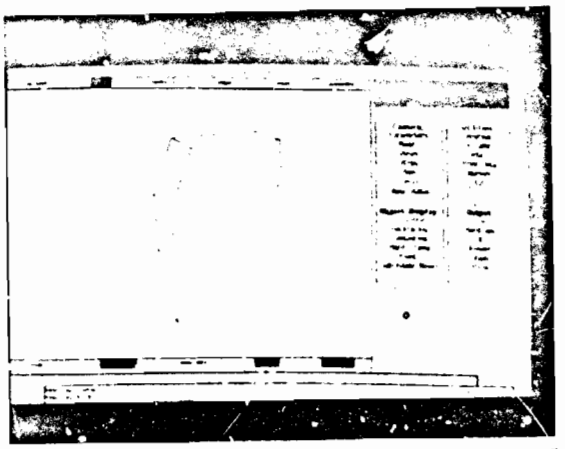


Figure 9. A 3-D model being prepared by the Symbolics S-Geometry solid modeling system.

or 3-D images of the object [Kanade 1983], [Faugeras 1984]. In this approach a 3-D model of the object is obtained by stereo matching and direct 3-D ranging. The description of the object is either a set of connected surfaces or a 3-D volume model. This segmentation is derived from the information in the scene. In the 2-D matcher described earlier, the model is simply selected line segments taken from an isolated view of the object.

Also under way are efforts to obtain 3-D object models from sensor data. The current experiments are directed at the extraction of a wire frame of an object by extracting curves of high-surface curvature in range data. These wire frames are then interpreted as solids by computing 2-cycle closures of space [Wesley and Markovsky 1982]. The information from several views can be combined by forming boolean intersections of the solid figures taken from each view.

THE ROLE OF GEOMETRIC REASONING

There are two central issues that arise from the procedures just described: 1) Specification of rules for matching object features across multiple perspective views; 2) Segmentation of object surface and volume into a compact and effective object representation. We propose that geometric reasoning using powerful algebraic deduction methods can significantly advance the state of the art in these two areas.

The formation of rules for matching features in stereo pairs or larger sets of multiple views is currently based on a heuristic analysis of the properties of the perspective image transformation and assumed properties of the objects in the scene such as polyhedra [Shafer et al. 1982].

An Algebraic Approach to Geometric Reasoning

We have been studying the application of a new method in geometric reasoning based on algebraic manipulation [Wu 1978]. The reasoning proceeds by manipulating a set of hypotheses that are represented as polynomials and attempts to establish the validity of a conclusion polynomial. The conclusion is shown to follow from the hypotheses if the conclusion can be expressed as a linear expansion in terms of the hypotheses.

A proof is established by dividing the conclusion polynomial by each of the hypotheses in turn. The remainder after the first division is used as the polynomial to be divided by the next hypothesis and so on. After all the hypotheses have been applied, the final remainder is checked. If the remainder is identically zero, then the conclusion is known to follow from the hypotheses.

If the final remainder is not zero, then it is not possible to conclude that the theorem is necessarily false. However, some very useful insights can be gained by determining additional conditions under which the final remainder does vanish. These conditions can be considered as additional hypotheses necessary to make the conclusion valid. Such conditions can provide useful insight into the problem under consideration. In many cases, the additional conditions correspond to unusual special cases that should not be overlooked in developing programs that use the geometric properties described in the theorem.

The procedure just described assumes that the hypotheses are in triangular form so that the variables are introduced one at a time in successive hypothesis polynomials. In symbolic form, if the variables are $[x_1, x_2, \cdots x_n]$, then the hypotheses should appear as

$h_1(x_1)$

$h_2(x_1, x_2)$

...

$h_n(x_1, x_2, \cdots x_n)$

INTERPRETING THE REMAINDER - AN EXAMPLE FROM VISION

An important aspect of the algebraic approach to theorem proving in geometry is that it is often possible to derive new conditions or constraints that were not obvious in the initial formulation of the problem. A mechanism for discovering new constraints is provided by examining both the degenerate conditions and the terms involved in any remainder. If a geometric interpretation can be made that causes the remainder to vanish, then this condition can be considered to be an additional hypothesis that is necessary for the theorem to be valid.

Likewise, the requirement that a degeneracy condition should never be zero, often imposes a new hypothesis for the validity of the theorem that was not understood in the original formulation. An example of this last case was observed in the parallel lines problem. In that example, it was implicitly assumed that points b and c were distinct. However such a condition must be an explicit hypothesis in order for the theorem to rigorously hold. These types of conditions often cause trouble in implementing programs to carry out geometric operations and relations. The process of discovering them in an examination of the proof of related theorems may be more efficient than exhaustive testing of the program.

To show the value of placing a geometric interpretation on the remainder, consider the problem illustrated in Figure 10. We are considering the standard arrangement of perspective viewing. The viewplane lies in the x-z plane and the viewing direction is along the positive z axis. The hypotheses are derived from the standard equations of perspective as well as the existence of two lines in the viewplane

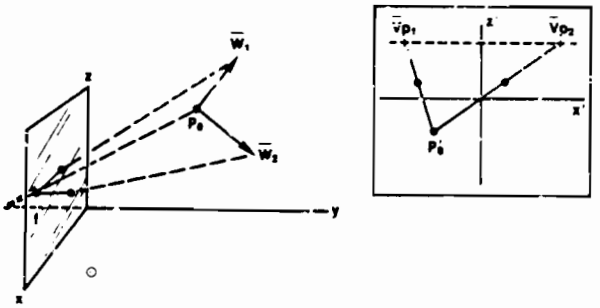


Figure 10. The perspective viewing of two lines in space. The horizon and vanishing points are shown in the box.

for which the vanishing point of each line has been given. In homogeneous coordinates, the vanishing point coordinate vector of a line, $\mathbf{Vp}$ in the viewplane can be related to its direction vector, $\mathbf{W}$, in space. the hypotheses are:

h_1: $W_{1x} Vp_{1z} - f W_{1x} = 0$

h_2: $W_{1y} Vp_{1z} - f W_{1z} = 0$

h_3: $W_{2x} Vp_{2z} - f W_{2x} = 0$

h_4: $W_{2y} Vp_{2z} - f W_{2z} = 0$

These equations express the location of each component of the vanishing points in terms of the direction of the lines, $\mathbf{W}_1$ and $\mathbf{W}_2$. The parameter, f, is the distance of the eyepoint from the viewplane. This distance is approximately equal to the lens focal length in a simple optical imaging system.

These hypotheses do not constrain the direction of the lines in space, but merely the dependence of the vanishing point locations on the line directions. Let us try to prove that the lines are, in fact, perpendicular in space. We have no reason to suppose that this theorem is valid, so we should expect to still have a remainder after dividing the conclusion by the hypotheses. The conclusion is

$g: W_{1x} W_{2x} + W_{1y} W_{2y} + W_{1z} W_{2z} = 0$

The dependent variables set is $[W_{2x}, W_{2y}, W_{1z}, *, *]$ With this choice, the hypotheses are already in triangular form. After synthetic division we obtain an expansion for g.

$-f^2 g = f W_{2x} h_4 + f W_{1y} h_3 + Vp_{2z} W_{2y} h_2 + Vp_{2x} W_{1x} h_1 - R$

The remainder R, after factoring is

$R = W_{1y} W_{2y} [f^2 + Vp_{1x} Vp_{2x} + Vp_{1z} Vp_{2z}]$

The degenerate condition requires that

$f^2 \neq 0$

This condition is an actual geometric degeneracy that corresponds to the viewpoint lying within the viewplane. In this case, the perspective transformation is ill-defined since all of space collapses into the viewpoint

Upon inspecting the remainder, several observations can be made. First, the remainder will vanish if either W_{1y} or W_{2y} are zero. These cases correspond to one or both of the lines being parallel to the viewplane. In such a case, the vanishing point of the line will be at infinity in the viewplane, and the perspective transformation equations become indeterminate. This case should be ruled out since it violates the assumptions underlying the hypotheses.

The second observation is related to the second factor in the remainder. In vector notation this can be rewritten as, $f^2 + Vp1 \bullet Vp2 = 0$

We have discovered a new hypothesis necessary to make the original theorem valid.[2] In order for two lines in a perspective image to correspond to perpendicular lines in space, this additional relationship between the vanishing points and the viewpoint distance must hold. This relation seems a useful constraint since it depends only on the focal length, f, of the image formation system, and the direction of the lines in the image plane. Several pairs of lines which satisfy the constraint for $f = 1$ are shown in Figure 11.

Current efforts are underway to provide a conceptual framework of geometric knowledge which can help direct and interpret the results of algebraic proof [Kapur et al. 1985]. The concepts are small groups of formal axioms that are part of standard geometry theory. The current issues under investigation are

- Representation and control for proof sequences.
- Inheritance mechanisms within the conceptual network.
- Techniques for embedding new theorems into existing concepts.
- Semi-automatic use of graphics to illustrate the theories.
- The integration of syntactic and logical operations

The final goal is to create a reasoning tool that can work cooperatively in the development of image understanding algorithms. We expect that the intial benefits will be the proof of correctness and an understanding of the limitations of the geometric assumptions and algorithms associated with perspective matching.

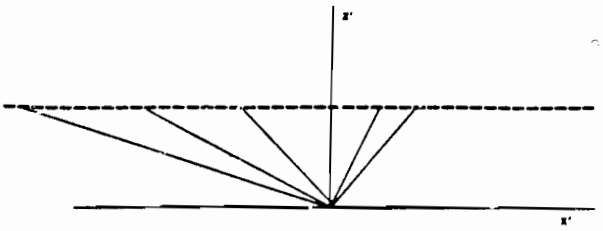


Figure 11. Three pairs of lines with vanishing points that satisfy the new constraint.

References

Asada, H. and Brady, J M., "The Curvature Primal Sketch," *Proc. Workshop on Computer Vision: Representation and Control*, 1984, p. 8.

Ayache, N., "A Model-Based Vision System to Identify and Locate Partially Visible Parts," *Proc. CVPR*, Washington, 1983.

Ballard, D.H., "Generalizing the Hough Transform to Detect Arbitray Shapes," *Pattern Recognition 13*, 1981, p. 111.

Brooks, R.A., "Symbolic Reasoning Among 3D Models and 2D Images," *Artificial Intelligence* 17, 1981, p. 285.

Canny, J., "Finding Lines and Edges," Report No. AI-TR-720, MIT Artificial Intelligence Laboratory, 1983.

Faugeras, O.D., "New Steps Toward a Flexible 3-D Vision System for Robotics," *Proc. 7th International Joint Conf. on Pattern Recognition*, 1984, p. 796.

Goad C., "Special Purpose Automatic Programming for Model Based Vision," Stanford University Report, ADP001198.

Grimson, W.E.L. and Lozano-Perez, T., "Search and Sensing Strategies for Recognition and Localization of Two and Three Dimensional Objects," *Proc. 3rd International Symposium on Robotics Research*, 1985.

Hermann, M., Kanade, T., and Kuroe, S., "Incremental Acquisition of a Three Dimensional Scene Model From Images," IEEE PAMI-6, 1984, p. 331.

Kapur, D., Mundy, J., Musser, D., Narendran, P., "Reasoning About 3D Space," *Proc. IEEE Conf. Robotics and Automation*, 1985, p. 405.

Markovsky, G. and Wesley, M.A., "Fleshing Out Wire Frames," *IBM J. Res. Dev. 24*, 1980, p. 582.

Mundy, J.L. and Joynson, R.E., "Automatic Visual Inspection Using Syntactic Analysis," *Proc. IEEE Conf. Pattern Recognition and Image Processing*, 1977, p. 144.

Shafer, S., Kanade, T., Kender, J., "Gradient Space Under Orthography and Perspective," *Computer Vision, Graphics and Image Processing 24*, 1983, p. 182.

Tomek, I., "Piecewise Linear Approximation," *IEEE Transactions on Computers C-23*, 1974, p. 445.

Porter, G.B and Mundy, J.L. , "A Model Driven Visual Inspection System," *First International Symposium on Robotics Research*, MIT Press, J.M. Brady and R.P. Paul ed., 1983.

Quam, L., "Image-Calc User's Manual," *SRI*, 1984.

Wu Wen-tsun, "On the Decision Problem and Mechanization of Theorem Proving in Elementary Geometry," *Scientia Sinica 21*, 1978, pp. 159-172.

2. Takeo Kanade indicated that this result has been published earlier. [Shafer et al. 1983]. The result was initially unknown to the author and derived independently in the manner described.

GC970145-14

STRUCTURE AND MOTION FROM IMAGES

J.K. Aggarwal
Amar Mitiche

Computer and Vision Research Center
The University of Texas at Austin
Austin, TX 78712

ABSTRACT

This paper reviews the research on structure and motion from images performed at the Computer and Vision Research Center of The University of Texas at Austin. Early work is briefly reviewed and the more recent work is described in greater detail. The recent work has focussed on developing methods which exploit explicitly the rigidity of objects in motion. Two methods are presented here. The first method relies on the observation of five points in two images, and uses the fact that distances between points of a rigid object do not change as a result of motion. The second method depends on the observation of four lines in three views and uses the fact that angles between the lines in a rigid configuration of lines do not change as a result of motion. In both cases, the transformation between views are computed and experimental results are presented.

1. INTRODUCTION

The perception and interpretation of motion of objects in space are among the fundamental capabilities of the human visual system. These capabilities have been and continue to be extensively investigated from the psychological and psychophysical viewpoints. More recently, computer vision has been concerned with the processing of sequences or collections of images with the objective of collecting information from the set as a whole that may not be obtained from any one image by itself. The study of cloud motion [1,2] may have been an early motivation that gave significant impetus to the analysis of motion by computers. However, at present time there are a multitude of applications driving the computer vision research into various facets of motion of objects in space. The application areas include medicine, autonomous navigation, tomography, communications and television, dancing and choreography, meteorology, robotics, animation and so on. Last, and certainly not the least, results in machine vision may contribute to the understanding of the functions of the human and animal visual systems. The broad interest is evidenced by the numerous workshops, special issues and conferences [3-16] almost exclusively devoted to analysis of time-varying imagery.

The present paper describes work done at the Computer and Vision Research Center of The University of Texas at Austin on the subject of motion. Early work is briefly surveyed and the more recent work is described in greater detail.

This research was supported in part by the Air Force Office of Scientific Research under Contract No. F49620-85-K-0007.

2. OVERVIEW

The study of motion by computer at the Computer and Vision Research Center has started with the work of Aggarwal and Duda [17] who proposed a mathematical model of cloud motion to use for the detection and measurement of this motion. The model consisted of planar rigid polygons. The motion of each polygon occurred in one of several parallel planes, and involved both translation and rotation as well, and could be correlated or independent. The objective of the study was to use a time-sequence of images of the polygons to compute their motions and to segment the scene into its component polygons. The number of planes and polygons were unknowns of the problem. The restriction of the figure shapes to polygons has been relaxed in a later studies in [18,19] to include curvilinear objects. An approach based on partial matching of contours in successive frames was developed to characterize motion and recognition of objects in successive frames.

The use of motion as a cue to image segmentation has been considered in [20] and [21]. A method is described in [20] where the images of moving objects are extracted by "subtracting" consecutive frames in a time-ordered sequence of frames and then focussing attention around "non-zero" areas of the "difference" frame; these non-zero areas indicated the presence of a moving object. An estimate/refine scheme was finally applied to extract the desired image regions (segmentation on the basis on motion). The study described in [21] used the idea of frame differencing coupled with a region growing operation, which started at the non-zero regions of the difference frame, to extract the image of moving objects.

All of the above mentioned studies are concerned with two-dimensional analysis of motion. Although the actual motions may be taking place in three-dimensional space, there was no attempt to recover or interpret them. The recovery and analysis of three-dimensional motion from two-dimensional images was considered by Roach and Aggarwal in [22,23]. Research described in [22] was concerned with the problem of determining the motion in space of planar-faced three-dimensional objects (blocks) from a sequence of (two-dimensional) images of these objects. The description of motions was qualitative and expressed as toward or away from the viewing system as well as left/right and up/down in the image plane. The strategy to recover such a description was to locate objects in the image and match them between frames. The image displacement of objects were computed from the centroid of their visible parts. Objects were segmented out by putting together detected faces using a heuristic scheme based on occlusion clues. Matching was multilevel, starting with the use of objects velocities and ending with local feature matching.

In [23] a quantitative evaluation of motion was sought. An approach was developed by which the position and motion of two or more points in space could be recovered from their observation in two distinct views. Camera imaging was modeled by central projection. Equations were derived from the projective relations which expressed the position of points in space in terms of the position of their projection in the image plane. The unknowns were the position of points in space with respect to one of the viewing systems, as well as the parameters of the relative displacement between these viewing systems. With six points, the proposed formulation yielded eighteen non-linear equations in eighteen unknowns when scale was fixed arbitrarily. The observation in [23] was that, unless a considerable number of points was used, the formulation was highly unstable numerically.

To overcome the problems associated with large systems of equations, a method is proposed in [24] which uses the usual projective relations and exploits a fundamental property of rigid objects; this property states that the distances between points of a rigid object do not change as a result of motion. The explicit use of this property leads to smaller system of equations which involve only the position of points in space and not the parameters of motion. But once the position of points in space is known, recovering the parameters of motion is a simple task. The approach, which relies on the observation of five points in two views, was shown to be numerically stable and well behaved.

Still using rigidity explicitly, an approach was developed in [25] which used the observation of lines instead of points. Rigidity was exploited by stating explicitly that the angular configuration between the lines of a rigid structure of lines does not change as a result of motion. The formulation used projective relations based on the observation of four lines in three views.

The study described in [26] was concerned with the interpretation of motion for jointed objects in addition to rigid objects. The main assumption was that motions are screws (translation + rotation) where the direction of the rotational axis remains fixed over short periods of time. This assumption was a generalization of that of planar motion considered in other studies which investigated the same problem ([27]). With this fixed axis assumption, the motion of any point on a rigid object relative to any other point on the same object is a circle in a plane normal to the fixed rotational axis. This circle projects as an ellipse on the image plane. The algorithm proposed in [26] exploited this constraint to interpret motion.

In the following sections, we will describe in greater detail the methods proposed in [24] and [25].

3. STRUCTURE AND MOTION FROM POINT CORRESPONDENCES

The problem of estimating the position and motion of an object in space from the observation of a small number of points in two distinct images of the object is considered in this section. In general, the solution consists of developing equations using projective relations for the observed points involving both the three-dimensional coordinates of the points and the parameters of motion. Then a typical counting argument dictates the number of points that must be observed in order to solve these relations. This usually has led to a large set of complicated non-linear transcendental equations. Linear methods were also considered when a larger number of points could be observed in the images.

The new notion in the method presented in the following is to use the principle of conservation of distances in rigid objects. This principle, which is the subject of a theorem in kinematics of solids, simply states an obvious fact: distances in a rigid configuration of points do not change during motion. It was shown in Mitiche [15,16] that this characterization of rigid motion can lead to powerful formulations of various structure and motion problems.

Consider the viewing system model shown in Figure 1. S_1 and S_2 represent the camera coordinate systems at two distinct viewpoints (obtained, say by a moving camera at two distinct instants of time or by two cameras from two different viewpoints). The approach is to write that distances between points of the rigid environment are the same whether expressed in S_1 or in S_2. There is no mention, at this stage, of the transformation that takes S_1 into S_2 or vice versa. Also, we use the scalar notation for projective relations. With this notation, each observed point contributes 2 variables (one for each coordinate system) and each pair of points gives one equation. If we take 5 points and set one variable arbitrarily to fix the global scale of the object, then we end up with 10 equations in 9 unknowns. Although this method does not provide an economy of points compared to some of the other methods ([23] for instance), we nevertheless arrive at a compact and robust formulation of the problem; each of the second order equations we obtain involves at most four of the nine position variables and none of the motion variables. As a result, the numerical solution of the system of equations is better behaved than it would otherwise be. Moreover, multiple solutions can only differ by a singular configuration or a reflection in space, once we fix the global scale factor.

After we determine the position of points, solving for the motion matrix is a simple matter of solving a 4 x 4 linear system of equations using 4 of the points. The actual parameters of motion can then be recovered analytically from the motion matrix.

3.1 Estimation of Object Position

Using the viewing system configuration shown in Figure 1, a point P_i in space with coordinates (X_i,Y_i,Z_i) in S_1 and (U_i,V_i,W_i) in S_2 is imaged on p_i on I_1 and q_i on I_2. Because P_i is on line $C_1 p_i$, there exists a real number $\lambda_i>1$ such that

$$X_i = \lambda_i x_i$$

$$Y_i = \lambda_i y_i \qquad (1)$$

$$Z_i = (1-\lambda_i) f$$

where (x_i,y_i) are the coordinates of p_i in the I_1-image

coordinate system and f is the focal length. Similarly, P_i is on line C_2q_i and if (u_i,v_i) are the coordinates of q_i in the I_2-coordinate system then there exist $\gamma_i > 1$ such that

$$U_i = \gamma_i u_i$$

$$V_i = \gamma_i v_i$$

$$W_i = (1-\gamma_i)f$$

The squared distance between points P_i and P_j expressed in S_1 is therefore

$$d_{ij}^2(S_1) = (X_i-X_j)^2 + (Y_i-Y_j)^2 + (Z_i-Z_j)^2$$

or

$$d_{ij}^2(S_1) = (\lambda_i x_i-\lambda_j x_j)^2 + (\lambda_i y_i-\lambda_j y_j)^2 + (\lambda_i-\lambda_j)^2 f^2$$

Similarly, the squared distance between P_i and P_j expressed in S_2 is

$$d_{ij}^2(S_2) = (\gamma_i u_i-\gamma_j u_j)^2 + (\gamma_i v_i-\gamma_j v_j)^2 + (\gamma_i-\gamma_j)^2 f^2$$

Now the principle of conservation of distance allows us to write (assuming, of course, identical units of measurement in S_1 and S_2):

$$d_{ij}^2(S_1) = d_{ij}^2(S_2)$$

or

$$(\lambda_i x_i-\lambda_j x_j)^2+(\lambda_i y_i-\lambda_j y_j)^2+(\lambda_i-\lambda_j)^2 f^2 = \qquad (2)$$

$$(\gamma_i u_i-\gamma_j u_j)^2+(\gamma_i v_i-\gamma_j v_j)^2+(\gamma_i-\gamma_j)^2 f^2$$

It may be seen that each point P_i contributes two unknowns, λ_i and γ_i, and each pair of points (P_i,P_j) gives one second order equation (Equation (2)). Therefore, 5 points yield 10 equations and 10 unknowns. The correspondence between points of the sets in the two views is, of course, assumed known. It may be noted that Equation (2) may be rewritten equivalently using a scale factor, a consequence of the fact that the scale of the observed structure of points cannot be recovered. We can fix this scale by fixing the distance of a point from one of the cameras which amounts to fixing arbitrarily one of the variables. Therefore, we end up with a system of 10 equations in 9 unknowns. It may be noted that each equation involves only 4 of the unknowns and that the formulation so far does not involve the parameters of the displacement between the two cameras. Because these parameters do not appear in the equations and also because only some of the unknowns of position appear in each of them, the resulting system of equations may be solved quite efficiently using existing numerical iterative algorithms.

3.2 Determining Motion

When the position of the points has been computed, determining the relative position of the cameras becomes a simple matter. Indeed, if one takes 4 non-coplanar points (4 of the 5 observed points in space or 3 of them with a fourth one generated using the product of vectors defined by these 3 points) and calling A_1 and A_2 the matrices of homogeneous coordinates of these in S_1 and S_2 respectively, the transformation (in homogeneous coordinate form) that takes S_1 onto S_2 is given as:

$$A_2 = MA_1 \qquad (3)$$

Since the 4 points are not coplanar then Equation (3) can be solved for M.

Now if we decompose the motion into a rotation through angle θ about an axis through the origin the direction cosines of which are n_1,n_2,n_3, followed by a translation (t_1,t_2,t_3) and if it is written as

$$M = \begin{bmatrix} a_1 & a_2 & a_3 & 0 \\ a_4 & a_5 & a_6 & 0 \\ a_7 & a_8 & a_9 & 0 \\ b_1 & b_2 & b_3 & 1 \end{bmatrix}$$

and one can show that

$$t_1 = b_1;\ t_2 = b_2;\ t_3 = b_3;$$

$$Cos\theta = \frac{a_1+a_5+a_9-1}{2}$$

$$Sin\theta = \frac{a_6-a_8}{2n_1}$$

$$n_1 = \sqrt{\frac{a_1-Cos\theta}{1-Cos\theta}};$$

$$n_2 = \frac{a_2+a_4}{2n_1(1-Cos\theta)};$$

$$n_3^2 = 1-n_1^2-n_2^2$$

Details of this derivation are found in [30] and [31].

3.3 Experimental Results

The algorithm for computing the position of points has been tested on synthetic data and camera-acquired pictures of real objects.

We used the FORTRAN subroutine LMDER [32] to solve the system of equations (2). This subroutine minimizes the sum of squares of m non-linear functions in n variables by a modification of the Levenberg-Marquardt algorithm. LMDER is an iterative procedure which requires initial guesses to the solution of the system of equations. Also, LMDER requires the Jacobian of the functions involved in this system. In our context the Jacobian can be calculated quite easily.

Two views of a jeep are shown in Figure 2a and 2b. The five points used are marked on the first view. The average error on the position of points for this typical example is 6% of the average distance between the points. For more details on this experiment, refer to [24]. Also refer to [24] for other examples on camera-acquired pictures and for statistical results on synthetic data.

4. INTERPRETATION OF STRUCTURE AND MOTION FROM LINE CORRESPONDENCES

In this section we discuss a solution to the problem of determining structure and motion from lines correspondences. More precisely, the problem is that of recovering the orientation and position of a set of lines in space from multiple views of these lines, as well as the relative displacement between the views. We consider undirected lines with no other cue such as points. Moreover, the lines are assumed to be in general position in space. The method is based on the observation of four lines in three distinct views. The case of two views has been shown to be inherently ambiguous [25]. Our method here exploits the principle of invariance of angular configuration with respect to rigid motion in addition to the usual projective constraints.

A viewing system is again represented by a central projection model. The geometric configuration with such a model is depicted in Figure 3. The projection center of the viewing system S is C and the image plane is I. The coordinate system in space is (O,X,Y,Z) where (O,X,Y) is the image coordinate system. The coordinates of C in S are $(0,0,f)$. The principal axis of the camera which this viewing system models, is aligned with the Z-axis. A point P in space projects on point p in the image as indicated in Figure 1 and line Δ would correspondingly project on line l.

When we use m views we will have m such viewing systems. With m views, we can write $m-1$ constraints on the orientation of lines for each possible pairing in the set of observed lines. Projective constraints are such that each line will contribute one unknown per view. With three views (the case we will treat in detail) for instance, four lines yield twelve equations in twelve unknowns which are solved for the orientation of lines in space. The rotational components of motion between the viewing systems are then readily recovered from these orientations. Finally, the translation components of motion (and therefore the position of lines in space) may be recovered.

4.1 Angular Invariance

The principle of angular invariance states that angles between lines of a set of lines in space do not change as a result of a rigid motion of this set of lines. In our investigation we consider undirected lines on which a sense cannot be fixed. Therefore, the constraint on angular invariance we exploit is one that preserves the quantity $\cos^2\theta$ where θ is any of the two angles θ_1 or θ_2 between two observed lines in space.

Let Δ_1 and Δ_2 be 2 lines in space the projection of which in S are respectively l_1 and l_2. Then, for any two non-zero vectors $\vec{v}_1$ on Δ_1 and $\vec{v}_2$ on Δ_2 we have:

$$\cos^2\theta = \left[\frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1|\,|\vec{v}_2|}\right]^2 \tag{4}$$

If we observe the same lines Δ_1 and Δ_2 in another reference system S', then one can write another expression for θ:

$$\cos^2\theta = \left[\frac{\vec{v}'_1 \cdot \vec{v}'_2}{|\vec{v}'_1|\,|\vec{v}'_2|}\right]^2$$

where $\vec{v}'_1$, $\vec{v}'_2$ have meanings similar to $\vec{v}_1$, $\vec{v}_2$.

Then the principle of invariance of angular configuration for Δ_1, Δ_2 between S and S' states that

$$\left[\frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1|\,|\vec{v}_2|}\right]^2 = \left[\frac{\vec{v}'_1 \cdot \vec{v}'_2}{|\vec{v}'_1|\,|\vec{v}'_2|}\right]^2$$

Now let P_1, P_2 be the end points of $\vec{v}_1$. P_1 and P_2 have, in this order, projections p_1, p_2 with image coordinates (x_1,y_1) and (x_2,y_2). Projective constraints, in their scalar form, allow us to write the following equations, where (X_1,Y_1,Z_1) and (X_2,Y_2,Z_2) are the coordinates of P_1 and P_2 respectively: there exists $\lambda_1 > 1$ such that

$$\begin{aligned} X_1 &= \lambda_1 x_1 \\ Y_1 &= \lambda_1 y_1 \\ Z_1 &= (1-\lambda_1) f \end{aligned} \tag{5}$$

Similarly, there is $\lambda_2 > 1$ such that

$$\begin{aligned} X_2 &= \lambda_2 x_2 \\ Y_2 &= \lambda_2 y_2 \\ Z_2 &= (1-\lambda_2) f \end{aligned} \tag{6}$$

The components of $\vec{v}_1$ are therefore:

$$V_{1x} = \lambda_2 x_2 - \lambda_1 x_1$$

$$V_{1y} = \lambda_2 y_2 - \lambda_1 y_1 \qquad (7)$$

$$V_{1z} = (\lambda_1 - \lambda_2) f$$

If we divide each expression in (7) by λ_1 and let $\lambda = \frac{\lambda_2}{\lambda_1}$ then the following unit vector, (A_1, A_2, A_3), is in the direction of $\vec{V}_1$

$$A_1 = \frac{\lambda x_2 - x_1}{a}$$

$$A_2 = \frac{\lambda y_2 - y_1}{a} \qquad (8)$$

$$A_3 = \frac{(1-\lambda) f}{a}$$

where $a = \sqrt{(\lambda x_2 - x_1)^2 + (\lambda y_2 - y_1)^2 + (1-\lambda)^2 f^2}$

If Q_1 and Q_2 are the endpoints of $\vec{V}_2$ with image coordinates (u_1, v_1) and (u_2, v_2) respectively, then it is straightforward to write expressions similar to those in (8) for $\vec{V}_2$:

There exists γ such that the vector (B_1, B_2, B_3) is a unit vector in the direction of $\vec{V}_2$

$$B_1 = \frac{\gamma u_2 - u_1}{b}$$

$$B_2 = \frac{\gamma v_2 - v_1}{b} \qquad (9)$$

$$B_3 = \frac{(1-\gamma) f}{b}$$

Where $b = \sqrt{(\gamma u_2 - u_1)^2 + (\gamma v_2 - v_1)^2 + (1-\gamma)^2 f^2}$.

Now we can write

$$\cos^2\theta = (A_1 B_1 + A_2 B_2 + A_3 B_3)^2 \qquad (10)$$

If we have another view of Δ_1 and Δ_2 where the viewing system of the second view is modelled by reference system S' then we can write expressions similar to those in (8), (9), and (10), i.e., there are λ' and γ' such that

$$\cos^2\theta = (A'_1 B'_1 + A'_2 B'_2 + A'_3 B'_3)^2 \qquad (11)$$

where

$$A'_1 = \frac{\lambda' x'_2 - x'_1}{a'}$$

$$A'_2 = \frac{\lambda' y'_2 - y'_1}{a'} \qquad (12)$$

$$A'_3 = \frac{(1-\lambda') f'}{a'}$$

and

$$B'_1 = \frac{\gamma' u'_2 - u'_1}{b'}$$

$$B'_2 = \frac{\gamma' v'_2 - v'_1}{b'} \qquad (13)$$

$$B'_3 = \frac{(1-\gamma') f'}{b'}$$

where (x'_1, y'_1) are the image coordinates of a point on line Δ_1; (x'_2, y'_2) are the image coordinates of another point on Δ_1; (u'_1, v'_1) are the image coordinates of a point on Δ_2; (u'_2, v'_2) are the image coordinates of another point on Δ_2. All these image coordinates are measured in the image plane of S'.

The principle of invariance of angular configuration for line Δ_1 and line Δ_2 between viewing systems S and S' is then written as

$$(A_1 B_1 + A_2 B_2 + A_3 B_3)^2 = (A'_1 B'_1 + A'_2 B'_2 + A'_3 B'_3)^2 \qquad (14)$$

or in expanded form

$$\frac{[(\lambda x_2 - x_1)(\gamma u_2 - u_1) + (\lambda y_2 - y_1)(\gamma v_2 - v_1) + (1-\lambda)(1-\gamma) f^2]^2}{a^2 b^2} \qquad (15)$$

$$= \frac{[(\lambda' x'_2 - x'_1)(\gamma' u'_2 - u'_1) + (\lambda' y'_2 - y'_1)(\gamma' v'_2 - v'_1) + (1-\lambda')(1-\gamma') f'^2]^2}{a'^2 b'^2}$$

Image coordinates involved in (15) are, of course, obtained from any two points on the images of line Δ_1 and any two points on the images of line Δ_2 in the image planes of S and S'. The unknowns are then λ, γ, λ', γ'; i.e., one unknown is involved per line and per view. We already have mentioned that the case of two views is highly ambiguous (refer to [25] for a proof).

If we consider four lines in space (therefore 6 possible different pairs of lines) and three views, then a simple counting argument determines 12 equations in 12 unknowns over

the three views. We obtain a system of equations with the following properties:

(a) The equations are of second order in each of the variables

(b) Only 4 of the 12 unknowns appear in each equation

(c) The motion parameters are not involved in the equations.

This system is solved for the orientation of lines in space. Because of the characteristics above, the system is better behaved numerically than it would otherwise be.

4.2 Determining Motion

In [25], it is shown that, once the orientations are known, the motion parameters as well as the exact position of the lines in space can be recovered by solving a linear system of equations. The derivations which are rather lengthy, are not reproduced in this paper. Full details are found in [25].

4.3 Experimental Results

We have generated randomly a large number of sets of 4 lines in space. We also added noise to the projection of these lines in the image plane. Two points on the image line are moved randomly in a 5x5 pixel area, creating a noisy image line. Results of the experiments with LMDER are shown graphically in Figure 4 which represent the average difference between computed and actual angles between lines in space versus precision of initial approximation. The observation is that LMDER performs reasonably well with fair initial approximation. Note that noise in the image is, by itself, responsible for approximately 2% error.

The performance of the method on real cases of camera-acquired pictures is currently under investigation.

5. SUMMARY

The detection and measurement of motion from images is of fundamental importance in a number of applications. Various approaches have been proposed and each of them, of course, has its advantages and limitations [33]. This paper has presented the ongoing work on the subject at the Computer and Vision Research Center at The University of Texas at Austin with emphasis put on the more recent results. These results exploit properties of rigid motion explicitly. Two such methods have been described. One of these methods is based on the use of points and exploits the fact that rigid motion does not change distances between points. The other method uses lines and the property that rigid motion does not alter angular configuration. Additional ongoing research on the recovery of structure and motion is based upon the observation of intensity and range images, presented in [34].

REFERENCES

[1] J.A. Leese, C.S. Novak, and V.R. Taylor, "The Determination of Cloud Pattern Motion from Geosynchronous Satellite Image Data," *Pattern Recognition*, 22, pp. 279-292, 1970.

[2] R.M. Endlich, D.E. Wolf, D.J. Hall, and A.E. Brain, "Use of a Pattern Recognition Technique of Satellite Photographs," *J. Appl. Met.*, 10, pp. 105-117, 1971.

[3] J.K. Aggarwal and N.I. Badler (Eds.), Abstracts of the Workshop on Computer Analysis of Time-Varying Imagery, University of Pennsylvania, Moore School of Electrical Engineering, Philadelphia, PA, April 1979.

[4] J.K. Aggarwal and N.I. Badler (Guest Eds.), Special Issue on Motion and Time-Varying Imagery, *IEEE Trans. on PAMI*, Vol. PAMI-2, No. 6, November 1980.

[5] W.E. Snyder (Guest Ed.), Computer Analysis of Time-Varying Images, *IEEE Computer*, Vol. 14, No. 8, August 1981.

[6] J.K. Aggarwal (Guest Ed.), Motion and Time Varying Imagery *Computer Vision, Graphics and Image Processing*, Vol. 21, Nos. 1 and 2, January, February 1983.

[7] T.S. Huang, *Image Sequence Analysis*, Springer-Verlag, New York, 1981.

[8] NATO Advanced Study Institute on Image Sequence Processing and Dynamic Scene Analysis, Advance Abstracts of Invited and Contributory Papers, June 21-July2, 1982, Braunlage, West Germany.

[9] Siggraph/Ciggart Interdisciplinary Workshop on Motion: Representation and Perception, Toronto, Canada, April 4-6 1983.

[10] International Workshop on Time-Varying Image Processing and Moving Object Recognition, Florence, Italy, May 1982.

[11] W.N. Martin and J.K. Aggarwal, "Dynamic Scene Analysis: A Survey," *Computer Graphics and Image Processing* 7, pp. 356-374, 1978.

[12] H.-H. Nagel, "Analysis Techniques for Image Sequences," in *Proc. IJCPR-78*, Kyoto, Japan, November 1978, pp. 186-211.

[13] J.K. Aggarwal and W.N. Martin, "Dynamic Scene Analysis," in the book *Image Sequence Processing and Dynamic Scene Analysis*, edited by T.S. Huang, Springer-Verlag, 1983, pp. 40-74.

[14] J.K. Aggarwal, "Three-Dimensional Description of Objects and Dynamic Scene Analysis," *Digital Image Analysis*, edited by S. Levialdi, Pitman, 1984, pp. 29-46.

[15] H.-H. Nagel, "What Can We Learn from Applications?" in the book *Image Sequence Analysis*, Edited by T.S. Huang, Springer-Verlag, 1981, pp. 19-228.

[16] T.S. Huang (Editor), *Image Sequence Processing and Dynamic Scene Analysis*, Proceedings of NATO Advanced Study Institute at Braunlage, West Germany,

Springer-Verlag, 1983.

[17] J.K. Aggarwal and R.O. Duda, "Computer Analysis of Moving Polygonal Images," *IEEE Trans. on Computers*, 24, No. 10, pp. 966-976, 1975.

[18] W.K. Chow and J.K. Aggarwal, "Computer Analysis of Planar Curvilinear Moving Images," *IEEE Trans. on Computers*, 26, No. 2, pp. 179-185, 1977.

[19] W.N. Martin and J.K. Aggarwal, "Computer Analysis of Dynamic Scenes Containing Curvilinear Figures," *Pattern Recognition*, 11, pp. 169-178, 1979.

[20] R. Jain, W.N. Martin, and J.K. Aggarwal, "Segmentation Through the Detection of Changes Due to Motion," *Computer Graphics and Image Processing*, 11, pp. 13-34, 1979.

[21] S. Yalamanchili, W.N. Martin, and J.K. Aggarwal, "Extraction of Moving Object Descriptions via Differencing," *Computer Graphics and Image Processing*, 18, pp. 188-201, 1982.

[22] J.W. Roach and J.K. Aggarwal, "Computer Tracking of Objects Moving in Space," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1, No. 2, pp. 127-135, 1979.

[23] J.W. Roach and J.K. Aggarwal, "Determining the Movement of Objects from a Sequence of Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2, No. 6, pp. 554-562, 1980.

[24] A. Mitiche, S. Seida, and J.K. Aggarwal, "Determining Position and Displacement in Space from Images," *Proc IEEE Conf on Computer Vision and Pattern Recognition*, San Francisco, CA, pp. 504-509, June 1985.

[25] A. Mitiche, S. Seida, and J.K. Aggarwal, "Interpretation of Structure and Motion from Line Correspondences," submitted to *Pattern Analysis and Machine Intelligence*.

[26] J.A. Webb and J.K. Aggarwal, "Structure from Motion of Rigid and Jointed Objects," *Artificial Intelligence*, 19, pp. 107-130, 1982.

[27] D.D. Hoffman and B. Flinchbaugh, "The Interpretation of Biological Motion, MIT AI Memo 608, Massachusetts Institute of Technology, Cambridge, MA 1980.

[28] A. Mitiche, "Computation of Optical Flow and Rigid Motion," *Proc. Workshop on Computer Vision: Representation and Control*, pp. 63-71, Annapolis, MD, 1984.

[29] A. Mitiche, "On Combining Stereopsis and Kineopsis for Space Perception," *Proc. First Conference on Artificial Intelligence Applications*, Denver, CO, pp. 156-160, 1984.

[30] R.Y. Tsai and T.S. Huang, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects With Curved Surfaces," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6, No. 1, pp. 13-26, 1984

[31] A. Mitiche and P. Bouthemy, "Tracking Modelled Structures Using Binocular Images," *Computer Vision, Graphics, and Image Processing*, to appear.

[32] LMDER, Minpack Subroutine, Argonne National Laboratories, March 1980.

[33] J.K. Aggarwal and A. Mitiche, "Structure and Motion from Images: Fact and Fiction," Third Workshop on Computer Vision: Representation and Control, Bellaire, MI, pp. 127-128, 1985.

[34] J.K. Aggarwal and M. Magee, "Determining Motion Parameters Using Intensity Guided Range Sensing," to appear in *Pattern Recognition*

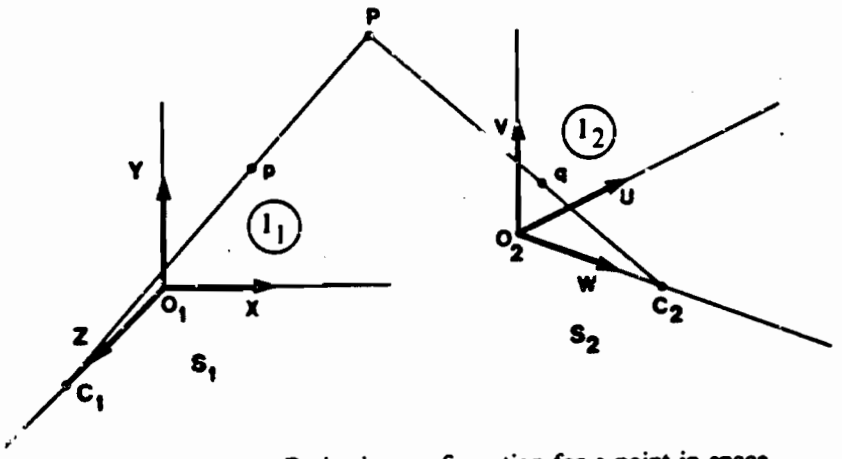
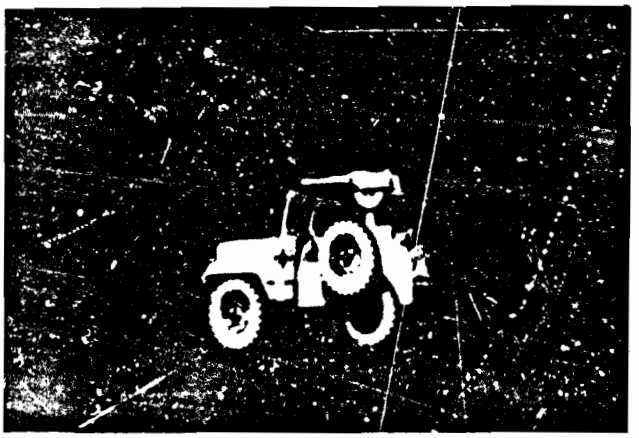
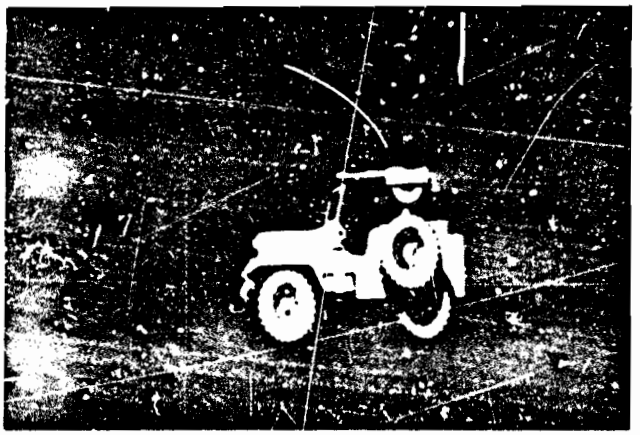


Figure 1. Projective configuration for a point in space with respect to viewing models S_1 and S_2.



(a)



(b)

Figure 2. One view of a jeep (a). Another view of the jeep (b). Selected points are marked on the first view (a).

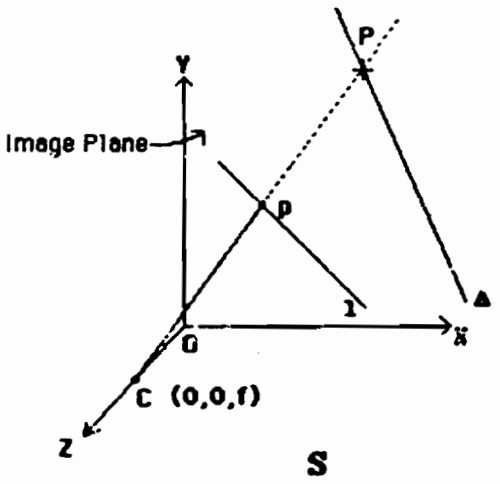


Figure 3. Projective configuration for a line in space with respect to viewing model *S*.

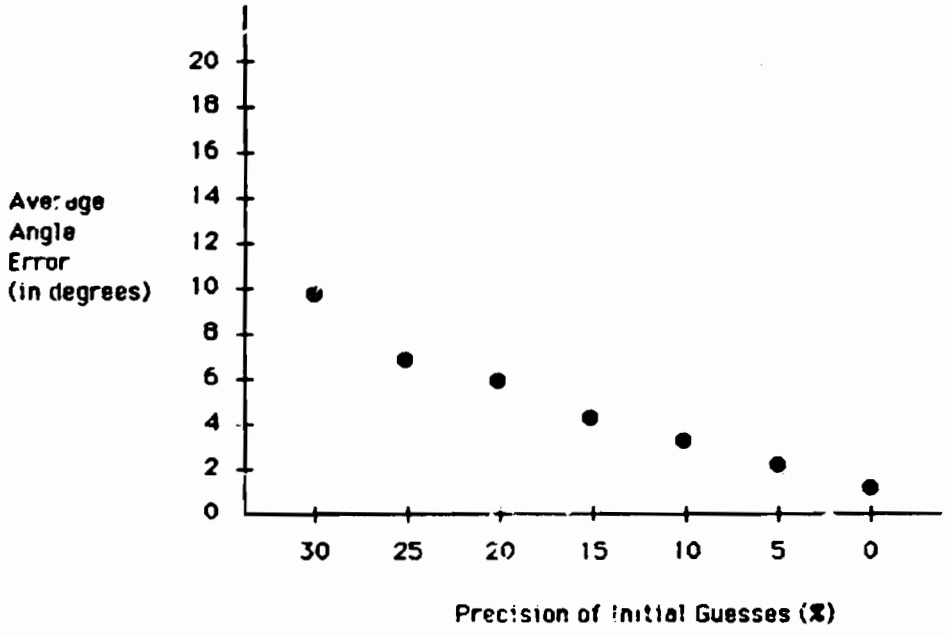


Figure 4. Average error in computed angles versus precision of initial guesses.

GC970145-15

SURFACES FROM STEREO

William Hoff
Narendra Ahuja

University of Illinois
Coordinated Science Laboratory
1101 W. Springfield Ave.
Urbana, IL 61801

ABSTRACT

This paper describes an algorithm and its implementation to compute the surface map of a scene from a stereo pair of images. The algorithm detects and matches zero crossings resulting after applying the Marr-Hildreth operator to the images. Ambiguities are resolved by choosing the matches which allow a smooth surface to be interpolated through depth points. It differs from other stereo algorithms in that it uses smoothness of the resulting algorithms as a criterion for matching. An interleaved sequence of matching-for-smoothness and surface interpolation operations generates a multiresolution hierarchy of surface maps, starting from the coarse and progressing towards the fine. As another important feature of the algorithm, at each stage the surface interpolation process takes into account the detected occluding and ridge contours in the scene, which are places where depth and orientation change abruptly. Interpolation is performed within regions enclosed by these contours. The algorithm takes a fairly unrestrictive view of real world objects captured in the following smoothness constraint: surfaces in the real world are smooth and continuous except across relatively rare occluding and ridge contours.

1. INTRODUCTION

This paper describes a stereo vision algorithm and its implementation. The purpose of stereo algorithms is to take two images of a scene, from slightly different viewpoints, and produce a complete depth map of the visible surfaces. The *usual* paradigm of these algorithms is: (1) *detect* suitable features in each image, (2) *match* corresponding features to determine their depths, and (3) *interpolate* to obtain a complete depth map [Barn82]. For the most part, the features that have been used in stereo have been low level - that is, they are only a function of the local intensities and are not semantic in nature. However, due to their simplicity, low level features can have many ambiguous matches, and this makes the matching step difficult. Also, occluding and ridge contours in the scene (where the depth and orientation, respectively, change abruptly) create difficulties for matching and interpolation.

The algorithm described addresses these difficulties by using a smoothness constraint for matching that explicitly incorporates the existence of depth and orientation contours in the computation, and enforces surface smoothness everywhere except across such contours. It integrates matching, contour detection and surface interpolation into a single process for extracting surfaces from stereo images. The algorithm is fairly domain-independent since it uses no constraint other than the assumption that objects have smooth surfaces, i.e., the depth and surface normal vary gradually except across relatively rare occluding and ridge contours.

The support of the National Science Foundation under grant ECS 8352408 is gratefully acknowledged.

2. BACKGROUND AND MOTIVATION

Existing algorithms may be classified according to the type of features they detect. Area-based algorithms attempt to match small windows in each image by correlating their intensities [Gean77, Hann80, Mora81, Pant78]. Edge-based algorithms detect edge or point-like features and attempt to match those [Arno78, Barn80, Grim81, Hend79]. Baker uses both areas and edges [Bake81]. Area-based algorithms have been applied successfully to the analysis of aerial images, where the terrain is smoothly varying and continuous. However, they have difficulty in dealing with scenes that contain depth and orientation discontinuities, because the matching windows may cross surface discontinuities. Edges are intrinsically more localizable and thus are more suitable in these cases. Also, they deal with intensity changes rather than raw intensities and thus are a better characteristic of physical changes in the scene.

Ambiguities may arise when matching edges because there is little to characterize an edge besides its location and orientation, and an edge in one image may match several in the other image. To select the correct match, Barnard and Thompson use the constraint that nearby image points should have nearly the same disparity [Barn80]. Baker selects edge matches so that the left-right ordering of edge points within a row in one image is the same as the left-right ordering in the corresponding row in the other image [Bake81]. In the algorithm of Marr-Poggio and Grimson [Grim81], ambiguities are resolved by choosing the match with a disparity that has its sign (convergent, divergent, or zero) the same as the majority of the non-ambiguous points in the neighborhood. Mayhew and Frisby note that edges which lie on a single continuous contour in one image should also have this continuity in the other image, and use this principle to eliminate ambiguities [Mayh81].

All previous algorithms complete the matching process *before* interpolating to obtain a dense depth map. Uniqueness of matching is only enforced by conditions that involve simple local relationships among disparity values as mentioned above and *not* the properties of the resulting surface. However, since a given disparity value implies a depth value, a stereo pair with matching ambiguities implies multiple surfaces, having different smoothness properties. The relative acceptability of these surfaces should be determined by the nature of the real world objects, namely, that their surfaces are smooth in the sense that the normal direction varies slowly, except across relatively rare creases and ridges. Thus, the expectations about surface characteristics have implications about resolution of matching ambiguities. This suggests that matching decisions should take into account the properties of the the resulting surfaces: the locations of depth and orientation contours as well as smoothness of the surface parts enclosed by these contours. This is in contrast to the traditional *first-finish-matching-then-interpolate* approach used by all existing stereo algorithms.

We conducted an experiment in human stereo vision to test the merit of our *smoothness-of-disparity* constraint, against the *constant-local-disparity* constraint used in the past [Hoff85]. We generated a random dot stereogram such that the use of the two different matching constraints would yield the perception of two different surfaces. The random dot stereogram portrays a surface whose height is that of a cosine wave, and which is unambiguous everywhere except for a small region centered on the peak. This ambiguous region can be perceived as a smooth continuation of the cosine wave, or as a surface which is locally rough but has approximately constant height. The observer fixates at the depth midway between the two surfaces. Figure 1 shows the experimental setup. In our experiment, most observers saw the smooth peak predicted by our smoothness constraint, instead of the rough peak predicted by the constant-local-disparity constraint.

Thus, to enforce the surface smoothness constraint it is not sufficient to use the local disparity histogram as has been done in the past. Rather, both the values of disparities as well as the locations of features giving rise to these values should be taken into account. The enforcement of local constancy of disparity biases the resulting surface towards a frontal orientation.

We have incorporated the more general, piecewise smoothness constraint into our algorithm to obtain *surfaces* from stereo. Note that as a consequence of the integration process, our algorithm more homogeneously carries out the stereo-to-surface transformation than the existing algorithms, where most of the computation is devoted to feature matching and thus a stereo-to-depth-points transformation. The surfaces in these algorithms only result from the final, independent step of interpolation through the depth points.

3. OVERVIEW OF THE ALGORITHM

An outline of the algorithm is shown in Figure 2. The processing is done in a coarse-to-fine-resolution mode. The algorithm starts with an initially specified, arbitrary estimate of the surface map, e.g., a flat frontal surface at some depth. At each resolution level, the following steps are performed. First, edges are detected in each of the two stereo images. Matches are sought for the edges in locations predicted by the surface depth at the previous, coarser resolution level. Each possible match obtained corresponds to a point whose position in the array as well as height are known. The match is recorded in a (x,y,z) array by locating points with appropriate height z for each edge point (x,y). This results in a sparse set of spikes with tips that must lie on the surface. Second, largest possible smooth patches are at centered at each (x,y) position in an image. Third, a comparison of adjacent patches identifies those pairs that differ in depth or orientation. Such pairs of patches yield estimates of depth and orientation contours in an image. Finally, a smooth surface is interpolated through depth points in each region surrounded by occluding or ridge contours. This gives a piecewise smooth surface map at the given resolution. The process is then repeated at finer resolution using the current surface to predict matching locations of edges at the finer resolution. Processing at successively finer resolutions yields surfaces at increasingly fine resolution.

The algorithm matches individual points in the left image with the corresponding points in the right image. Currently, these points are the zero crossings of the Marr-Hildreth operator. For each pair of corresponding points, the depth may be calculated from the disparity in the positions of the two points. The matching is driven from left to right, so that the result is a set of points in the left image, each labeled with one or more depth values.

The depth points have the following characteristics: First, they may have ambiguous depth values. This is caused by the fact that in some cases, a point in the left image can match more than one point in the right image, implying more than one possible depth for that point. Second, some of the points may have no correct depth value, due to noise or occlusion. There may also be incorrect guidance for the matching from the coarser levels. Third, the points are sparse, which is characteristic of zero crossings. Fourth, the depth values are noisy, which can be caused by image noise. It can also be caused by the blurring effect of the Marr-Hildreth operator. In general, the uncertainty in the position of the zero crossing is proportional to the size of the operator.

Thus, to solve the matching problem and interpolate a surface, we invoke the smoothness constraint -- that objects in the real world tend to have piecewise smooth surfaces. Thus, the depth values are assumed to be noisy samples of a surface which is smooth in the sense that the depth and surface normal vary slowly, but which may contain depth and orientation discontinuities at the relatively rare occluding boundaries and creases. We start with reconstructing a polyhedral approximation of the original surface by fitting planar patches to the depth points. From these, we obtain a piecewise smooth approximation to the original surface, with depth and discontinuities located.

4. DETAILED DESCRIPTION

The following is a detailed description of the algorithm. It was implemented in "C" on a VAX 11/780. Some of the runs were done on a Gould 9050 superminicomputer.

4.1. DETECTION OF FEATURES

The first phase of the algorithm is similar to the Marr and Grimson method of detecting features. The left and right images are each convolved with the Marr Hildreth operator (Laplacian of a Gaussian) of different sizes (different widths of the Gaussian). Zero crossings are then detected. These correspond to significant intensity changes in the image. The result is a set of left/right pairs of images showing zero crossing locations. Each pair is at a different scale of resolution. In the implementation the resolution was reduced by a factor of two at each level, yielding image sizes of 256x256, 128x128, and 64x64. The effective width of the Marr-Hildreth operator (the diameter of the central negative region) was the same for each level, *i.e.*, 6.

4.2. MATCHING ZERO CROSSINGS

Matching of zero crossings is done in a coarse to fine process. At the coarsest level the algorithm must be supplied with an initial estimate of the depth map, *i.e.*, a constant. To match a zero crossing in the left image, the algorithm attempts to find one or more similar zero crossings in the right image. It searches for candidate zero crossings in a small horizontal interval (window) centered at the location predicted by the depth estimate. The epipolar lines are assumed to be horizontal, so that searching is restricted to one dimension. The window width is equal to twice the width of the Gaussian filter used to create the zero crossings at this level, as in the Marr-Grimson algorithm. This has the consequence that there is a 50% chance that there will be only one zero crossing of the correct sign in the window, otherwise there will usually only be two. A zero crossing that is found in the window is a match if it has a orientation similar to the orientation of the zero crossing in the left image. In the implementation orientations are quantized into 32 quantizations, corresponding to 0-360 degrees, and a difference of up to 2 quantizations is allowed for a match. If there is one match, the point is unambiguous and the depth (or disparity) is known. If there is no match, nothing further can be done with this point and so it is ignored. If there are two or more matches, the point

is ambiguous and the algorithm tries to resolve this ambiguity in later processing.

The program attempts to match only non-horizontal zero crossings, since the disparity of horizontal zero crossings is subject to large error. The percentage of ambiguous points is smaller than 50%, due to the added constraint that the orientations must be similar. Also, for computational reasons, the maximum number of matches that are allowed is two. If a point has more than two matches, it is treated as unmatchable.

A surface is now interpolated through the depth values by the interpolation process described below. To match the next finer level of zero crossings, the depth estimate is given by the depth[1] of the surface at the same point in the current level. Thus, the interpolated surface at a given level guides the matching at the next level.

4.3. FITTING PLANES

The algorithm fits planes to the depth values in circular regions centered at every 4th grid point in the image. The largest possible disc is identified at each point under the constraint that the depth points in the disc are a good fit to a plane. We used a maximum radius of 20 to limit the computation. For each region, the two planes having the smallest squared error of fit to the depth values in that region are found. There may be more than one plane because some of the depth values may be suspected to be mismatches and may be ignored in fitting a patch, while other depth points may be ambiguous, with multiple values.

To determine whether or not the points are a good fit to a plane requires an estimate of the noise in the depth values. We assume that the major component of this noise is due to the fluctuation of the zero crossings about the true edge position. Berzins [Berz84] did some analysis on the displacement error for specific image situations, and found that the error was usually much less than σ_G, where σ_G is the standard deviation of the Gaussian in the Marr-Hildreth operator:

$$\nabla^2 G(r) = -\frac{1}{\pi\sigma_G^4}\left[1-\frac{r^2}{2\sigma_G^2}\right]\exp(-r^2/2\sigma_G^2)$$

Even in unusually bad cases, the error was comparable to σ_G. If the displacement error is normally distributed, then 95% of the time the error will be less than $2\sigma_N$, where σ_N is the standard deviation of the noise. Therefore, we assume $2\sigma_N = \sigma_G$, which is consistent with the displacement of zero crossings observed in our experiments (see Section 5). For a given number of points in a region, within the distance $2\sigma_N$ of the plane, the probability of the sum of squared errors being less than or equal to ϵ^2 may be determined from the chi-squared distribution. The program determines the maximum expected squared error for a 95% confidence level. If the squared error exceeds this value, the plane is rejected. As a further check on the validity of the planar fit for the region, we count the number of points that lie beyond the distance $2\sigma_N$ from the plane. If this number exceeds the expected 5% level, the plane is rejected.

A crucial part of this algorithm is the use of the Hough transform to fit planes. This is important because the ambiguous and mismatched points would lead to combinatorial explosion if a standard least squared method such as Gaussian elimination was used to fit planes to each possible subset of depth points in a region. The Hough transform is used to calculate the least squared fit plane by the following method: A three-dimensional parameter space is set up, with each dimension corresponding to a parameter in the equation of the plane:

$$z = ax + by + c$$

For each point p_i in the region, the parameter space cells are incremented at the locations corresponding to the solutions (a, b, c) of the equation

$$c = z_i - ax_i - by_i + \epsilon_i$$

where ϵ_i represents the amount of error of fit of the point (x_i, y_i, z_i) to the plane represented by (a, b, c). The array is incremented at each location (a, b, c) by the amount ϵ_i^2 – the squared error of fit of (x_i, y_i, z_i) to the plane (a, b, c). After all points have been entered in this manner, the minimum entry in the parameter array represents the solution with the minimum squared error.

To allow for mismatched points and ambiguous points, a maximum allowed distance of a point from the plane is computed. In the implementation, twice the estimated standard deviation of the noise was used as the maximum distance. If a point is further than this from the plane, it is considered to be an outlier to that plane, and its squared error does not contribute to the total. In the case of ambiguous points, only one of the depth values contributes to any plane - the one which is closest in depth. An important advantage of the Hough transform is that it requires a constant amount of work for each depth value, and the amount of work is not dependent on the number of ambiguous points or mismatches.

A disadvantage of the Hough transform method is the limited resolution of the parameter space. In the implementation, the parameter space was 11x11x16, with the first two dimensions used for the x and y slopes from -1.0 to 1.0, and the third dimension for the z offset. This allowed a resolution of 0.2 in the slope, and 1.0 in the z value. A higher resolution could be used, but at the cost of additional computation. However, because the planes obtained are only local approximations, a very fine resolution is not crucial.

4.4. CONFLICT RESOLUTION

Overlapping planar patches obtained from the previous step may be inconsistent, in the sense that a depth value that is a good fit to one of the patches is treated as an outlier by the other patch. This situation occurs, for example, at an occluding boundary, where a patch from one side of the boundary may extend for a short distance into the other side, treating points from the other side as outliers. These patches are shrunk until they no longer contain outliers. If an image region has multiple candidate patches, the largest of the shrunken patches is selected as the best fit for the region. If there is no one patch that is largest, the patch is selected that is most consistent with its neighbors, or failing that, the one with the smallest least squared error.

If an outlier point is really a mismatch, it should be ignored and any patch which contains it should not be reduced. This situation can usually be detected because the planar patches which contain the outlier point will generally be much smaller than the ones that do not, due to the fact that the outlier point is not part of any consistent surface. The algorithm handles this situation by keeping track of which patches are *maximal*, i.e. not completely contained in any other patch. If an outlier point is not on any maximal patch, then it is probably a mismatch and can be ignored. When a patch is reduced in size, other patches within it may become maximal. Therefore, this process is repeated until no more shrinking is necessary.

The result is a set of planar patches which are no longer inconsistent, and there is a unique patch for every region. The ambiguous depth values can now be resolved by choosing the match which is on the largest number of maximal planes. If none of the depth values for a point are on any maximal plane,

[1] In this paper, "depth" and "disparity" are used interchangeably because the depth can be readily calculated from the disparity and the camera model.

point is a mismatch and is removed.

4.5. LOCATING EDGES

The next step in the algorithm is to locate depth discontinuities. Such discontinuities should ideally be defined by adjacent pairs of patches that differ in depth. However, due to the sparsity of the input data points, it is possible that there will be planar patches with good fits across discontinuities. These patches will generally be small compared to the patches which do not cross the discontinuity. Therefore, the edge can be detected by examining the larger patches first, and eliminating those smaller patches which cross the edge.

The program checks the difference in depth, at the midpoint of overlap, for all overlapping or adjacent patches. If this difference is greater than a threshold[2] then an edge point is marked there. The edge is grown for a short distance perpendicular to the line joining the centers of the overlapping patches, such that its extension does not contradict the depth points in the neighborhood -- meaning that the edge consistently separates high depth values on one side and low depth values on the other. Any smaller patches which overlap this edge band are shrunk so that they do not contain the band.

The result is a set of possible edge points, which indicate where the occluding contours could lie. The edge points are thinned until a one-pixel wide contour is obtained. Any patches which overlap this contour are shrunk.

After the occluding contours are found, the orientation discontinuities could then be found by the same methods - *i.e.*, checking overlapping or adjacent patches for significant orientation differences at the midpoint of overlap. This step is not implemented in the current version, but will be in future versions.

4.6. GENERATING A COMPLETE SURFACE MAP

The final step is to interpolate to obtain a complete surface. The algorithm of Terzopoulos [Terz83] was used. The input to this algorithm is the unambiguous depth points and the estimated depth discontinuities. We also provided an initial surface estimate, which was obtained by just averaging the heights of the planar patches. The algorithm iteratively smoothed the surface while requiring it to pass close to the depth points.

After iterating until the amount of change was very small, the surface was analyzed for evidence of depth discontinuities that were not detected earlier. If the surface gradient is high, then there probably should be a depth discontinuity there. These were located by taking the Laplacian of the surface, and noting the zero crossings. At zero crossings where the surface gradient exceeded a threshold, a new edge point was marked. The Terzopoulos algorithm was again run with the new edges until change in the surface was small. The result is the final estimate of the surface at the current level.

This surface is now used as the depth estimate for matching at the next level. In the vicinity of occluding contours, two depth estimates are used -- one from the high side of the surface, and one from the low side. This is done because the exact location of the occluding contour is uncertain.

5. EXAMPLES

Results are presented for running the algorithm on the stereo pair of images shown in Figure 3. These images show a baseball resting on a newspaper. Because of the aspect ratio of the cameras, the spherical baseball appears to be an ellipsoid. Each image is 256x256 pixels in size. The images were separately convolved with the Marr-Hildreth operator at different scales, and zero crossings extracted. Figure 4 shows the zero crossings at three scales of resolution: 256x256, 128x128, and 64x64. The effective width of the operator is 6 for each case.

5.1. RESULTS FROM BASEBALL IMAGES

A constant depth estimate of 3 was supplied to the algorithm to match the 64x64 level. The estimate corresponds to a depth midway between the top of the ball and the newspaper[3]. The initial matching results for this level are summarized in Table 1, as are the matching results for the other levels.

Table 1
Matching Summary

	Size		
	64	128	256
0 matches	32	230	1799
1 match	221	1328	5169
2 matches	21	269	798
> 2 matches	4	19	87
Total non-horizontal zero crossings	278	1846	7853

Planar patches were fit to the depth points obtained from the above matching process as described earlier. The ambiguous depth points were resolved and mismatched points were removed. The algorithm then searched for depth edges by comparing the heights of adjacent and overlapping patches. No depth edges were found at this level.

The Terzopoulos surface interpolation algorithm was then run, using the unambiguous depth points as depth constraints. After stabilization, surface points of high gradient were found and marked as depth edges. These new depth edges are shown in Figure 5. Surface interpolation was done again, but smoothing was not done across the new depth edges. The final surface for this level is shown in Figure 6. A contour plot is shown in Figure 7.

The entire process was repeated at the 128x128 level, but using the surface obtained at the 64x64 level as a disparity estimate for matching. The 128x128 level is different from the 64x64 level since edges were detected by comparing adjacent planar patches. After the Terzopoulos algorithm was run, these edges were modified and extended. Figure 8 shows the final depth edges obtained by the algorithm for this level. Figure 9 shows the surface as a height field, and Figure 10 shows the surface as a contour plot. Note that the height of the surface is double that of the 64x64 level. This is because the disparities have doubled, due to the increased resolution.

The process was again repeated at the 256x256 level, using the surface obtained at the 128x128 level as a disparity estimate. Figure 11 shows the final depth edges obtained by the algorithm for this level. Figure 12 shows the surface as a height field, and Figure 13 shows the surface as a contour plot.

5.2. COMPARISON WITH GROUND TRUTH

The images were taken using cameras looking down from the height of approximately 40". The cameras were separated by a distance of approximately 12", in an epipolar configuration.

[2] The threshold in the implementation was taken to be $3.5\sigma_y$.

[3] A wide range of values for the estimate will work, because the width of the matching window is much larger than the range of disparities for this scene.

The baseball has a diameter of about 3". Although these parameters were not measured accurately, we measured the disparity at certain image points to obtain the ground truth. We located distinctive markings in both images and noted the difference in their positions. The disparities measured in this way were found to be roughly constant at about 8 for the newspaper and about 20 at the top of the baseball[2]. The lateral dimensions (in pixels) of the baseball image were also measured by hand. The true surface map was calculated by using the equation of an ellipsoid for the ball and a plane for the newspaper.

The calculated disparities at zero crossings were compared to the ideal disparities at the same locations. The comparison is shown in Table 2. Figures for each level in Table 2a are for the initial matches for that level, obtained by a search in the vicinity of the positions predicted by the coarser level (or constant if initial) surface estimate. If a zero crossing match is ambiguous, the closest depth value to the ideal was taken. The data shows that the assumption made about the magnitude of the disperity noise was reasonable. Since the width of the filter for all three levels was 6,

$$\sigma_G = \frac{w}{2\sqrt{2}} = 2.12$$

is the standard deviation of the Gaussian, and we have assumed that the standard deviation of the noise was

$$\sigma_N = (1/2)^* \sigma_G = 1.06$$

In each of the cases, about 95% of the errors were less than 2 pixels, which is consistent with the value of σ_N. The data also shows that roughly 5% of the points have large errors and should be treated as mismatches.

Figures for each level in Table 2b are for the zero crossings remaining after resolving ambiguities and removing mismatches. The data shows that most of the zero crossings that were removed were points that had large errors. Most of the remaining points with large errors lie near the contour of the ball. Figure 14 shows the locations of the remaining points with 3 or more pixels of disparity error, for the 256x256 level.

The ideal surface for the 256x256 level is shown as a height field in Figure 15, and a contour map in Figure 16. The ideal surface agrees with the surface obtained to within 1 pixel in most places.

5.3. COMMENTS ON PERFORMANCE

The depth edges detected by the algorithm are occasionally missing or misplaced. In the future, we plan to incorporate a 3D edge smoothness constraint so that they form smooth contours.

The surface interpolation step using the Terzopoulos algorithm is probably unnecessary, because the surface information is already available locally in the form of the planar patches. In fact, a good approximation to the final result is obtained by just averaging the heights of the overlapping patches at each point, and this approximation is used as the starting surface for the Terzopoulos algorithm. However, a more sophisticated method of combining the overlapping patches appears necessary to obtain the final surface, because the surface obtained by just averaging looks blocky.

The algorithm occasionally has trouble identifying mismatched points near the border of the image. This happens whenever the error points do not have enough correct points nearby. Any error points near the image border are less likely to be corrected since they have fewer surrounding points and hence fewer surrounding correct points. Therefore, mismatched points are more likely to survive near the image border than away from the border. This is observable at the right border of the 256x256 surface, where a few points with incorrect depth values cause the surface to rise sharply. This problem will be addressed in the future.

[2] Actually, the disparity of the newspaper rises gradually from about 7.5 at the top of the image to about 9.5 at the bottom. The gradual rise can be observed in the reconstructed surface, particularly at the 256x256 level.

Table 2
Accuracy of Disparities at Zero Crossings

Table 2a: Initial matches suggested by coarser level.

		64	128	256
error	0 - 1	211 (87%)	1438 (90%)	4889 (82%)
	1 - 2	20 (8%)	91 (6%)	643 (11%)
	2 - 3	1 (0%)	7 (0%)	67 (1%)
	3 - 4	5 (2%)	2 (0%)	86 (1%)
	> 4	5 (2%)	49 (3%)	282 (5%)
Totals		242 (100%)	1597 (100%)	5967 (100%)

Table 2b: Final matches after surface-based processing.

		64	128	256
error	0 - 1	210 (89%)	1431 (93%)	4834 (87%)
	1 - 2	21 (9%)	101 (7%)	631 (11%)
	2 - 3	1 (0%)	6 (0%)	51 (1%)
	3 - 4	5 (2%)	1 (0%)	15 (0%)
	> 4	0 (0%)	2 (0%)	40 (1%)
Total		237 (100%)	1541 (100%)	5571 (100%)

6. CONCLUSIONS

The following are some salient features of our approach to stereo.

Integration of Matching and Interpolation

The most novel characteristic of our approach is the use of the surface smoothness criterion for stereo matching, and thus an integration of matching and surface interpolation operations. This is in contrast with the traditional sequential ordering of matching *followed* by interpolation. The control passes back and forth between matching and interpolation processes, each depending on the result of the other to make progress, and generating a progressively refined set of depth maps of a scene at increasing degree of resolution. A given coarse level surface predicts the locations of edge matches at the next finer level. The matched features at the finer level provide a more refined surface which in turn predicts pairs of edges to be matched at the next finer level of resolution.

Occluding and Ridge Contours

Another important characteristic of our approach is that our smoothness constraint explicitly incorporates the existence of depth and orientation discontinuities in the computation. It is fairly domain-independent, *i.e.*, it uses no constraint other than the assumption that objects in the real world tend to have smooth surfaces, *i.e.*, the depth varies gradually except across relatively rare, occluding and ridge contours. These contours are *constantly* detected and a smooth surface is interpolated allowing depth/slope discontinuity across the contours, thus implement-

g the *piecewise-smooth* model of real world objects. In our rrent implementation, we detect only occluding contours. nis causes no problems with the baseball image. We are rrently incorporating the detection of ridge contours in our gorithm.

There is an important positive side effect of explicit detection of contours. We can identify the regions corresponding to e parts of the scene not visible from each camera. In case of the seball image these are the regions immediately to the left of e left occluding border, and to the right of the right occluding rder. The mismatches in the these regions can now be avoided. e plan to incorporate this feature in our algorithm.

ntinuity of Discontinuities

Finally, our approach enforces smoothness and continuity 3-D occluding and ridge contours. This constraint is based on the assumption that real world objects have surfaces that ve smooth borders. We have not incorporated such 3D noothing in our current implementation. Nor have we done aborate 2D extension of edge segments detected from adjacent tches. These are other additions we plan to make to our rrent implementation.

Our current implementation of plane fitting to depth nts, and edge detection from the resulting patches is preliminary and needs refinement. The poor performance of these steps sults in the errors observed near image borders, near the baseall borders, and the poor quality of contours. We expect to nprove this performance. Once we have more accurate patches, e plan to perform the surface interpolation by combining the rious planar patches at each point, and not use the global terpolation performed by the Terzopoulos algorithm.

One final comment before we close. The baseball image ve we have chosen to run our algorithm on presents a particularly harsh test of the algorithm near the baseball border. This is cause large surface steepness and occluding contours occur at e same locations. This makes the need for better planar patch lection and edge detection even more important.

References

[Arno78]
Arnold, D., "Local context in matching edges for stereo vision," *Proc. Image Understanding Workshop*, May 1978.

[Bake81]
Baker, H.H., *Depth from Edge and Intensity Based Stereo*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1981.

[Barn80]
Barnard, S.T. and W.B. Thompson, "Disparity analysis of images," *IEEE Trans. Pattern Anal. Machine Intell.*, July 1980.

[Barn82]
Barnard, S.T. and M.A. Fischler, "Computational Stereo," *Computing Surveys*, vol. 14, no. 4, December 1982.

[Berz84]
Berzins, Valdis, "Accuracy of Laplacian Edge Detectors," *CVGIP*, vol. 27, pp. 195-210, 1984.

[Genn77]
Gennery, D., "A stereo vision system for an autonomous vehicle," *IJCPR*, 1977, p. 576.

[Grim81]
Grimson, E., *From Images to Surfaces*, MIT Press, Cambridge, Massachusetts, 1981.

[Hann80]
Hannah, M.J., "Bootstrap Stereo," *Proc. Image Understanding Workshop*, College Park, Md, April 1980.

[Hend79]
Henderson, R.L., *et al.*, "Automatic stereo reconstruction of man-made targets," *Soc. P.I.E.*, vol. 186, no. 6, 1979.

[Hoff85]
Hoff, W.A., and N. Ahuja, "Depth from Stereo," *Fourth Scandinavian Conference on Image Analysis*, June 18-20, 1985, Trondheim, Norway, 761-768.

[Mayh81]
Mayhew, J.E.W., and J.P. Frisby, "Psychophysical and Computational Studies towards a Theory of Human Stereopsis," *Artificial Intelligence 17*, 1981, pp. 349-385.

[Mora81]
Moravec, H., "Rover visual obstacle avoidance," *Proc. 7th IJCAI*, August 1981, pp. 785-790.

[Pant78]
Panton, D.J., "A flexible approach to digital stereo mapping," *Photogramm. Eng. Remote Sensing 44, 12*, Dec 1978, pp. 1499-1512.

[Terz83]
Terzopoulos, Demetri, "The role of constraints and discontinuities in visible-surface reconstruction," *Proc. IJCAI*, pp. 1073-1077, Karlsruhe, August 1983.

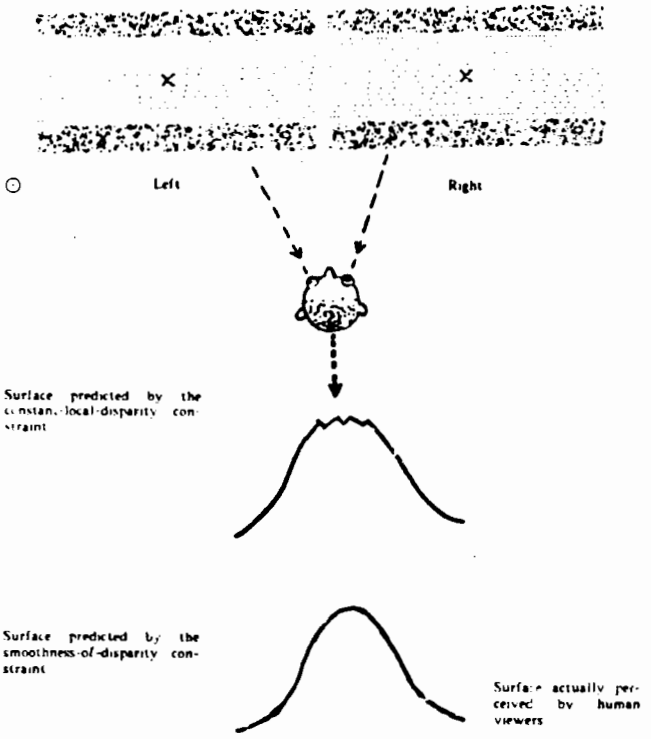


Figure 1.

An experiment with human stereo vision to contrast our surface smoothness constraint for stereo matching against the traditional constant-local-disparity constraint. An ambiguous random dot stereogram is shown at the top and the two surfaces predicted by the constraints are shown below. Most observers see the bottom surface.

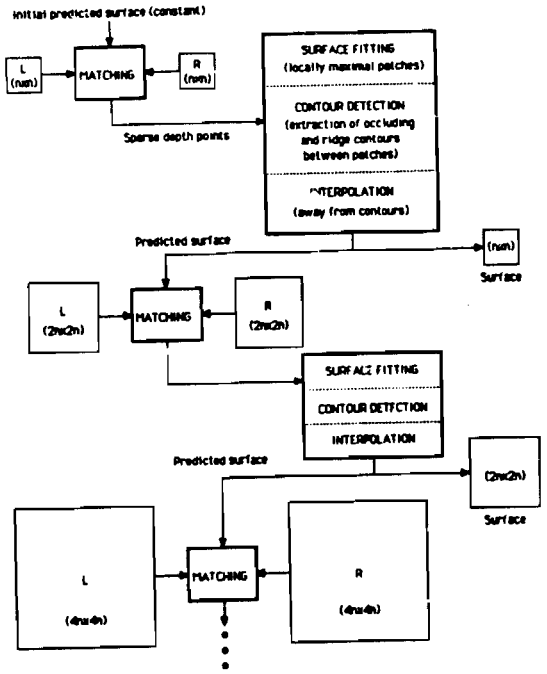


Figure 2.

A schematic view of control flow and computation in our proposed stereo algorithm.

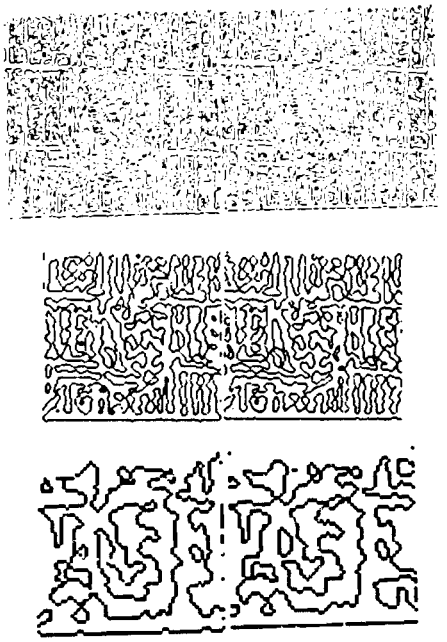


Figure 4.

Zero crossings for the 64x64, 128x128, and 256x256 levels of resolution.

Figure 3. Original stereo pair of images.

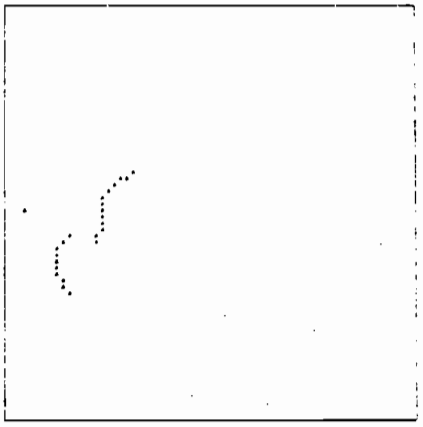


Figure 5. Depth edges detected at the 61x64 level.

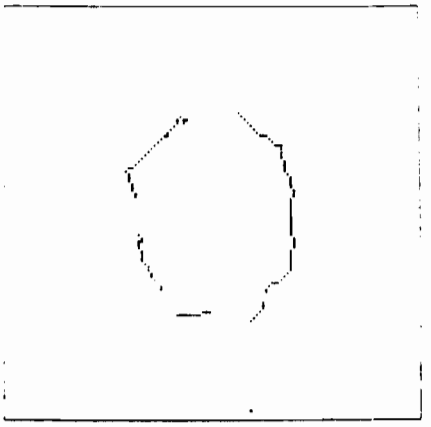


Figure 8. Depth edges detected at the 128x128 level.

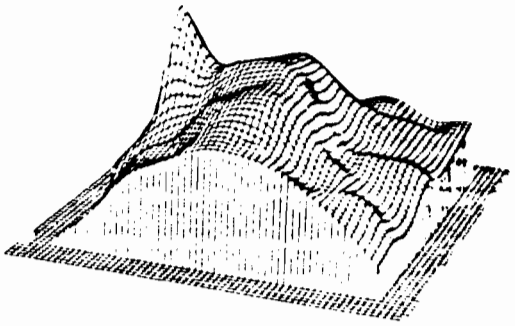


Figure 6. Final surface for the 64x64 level.

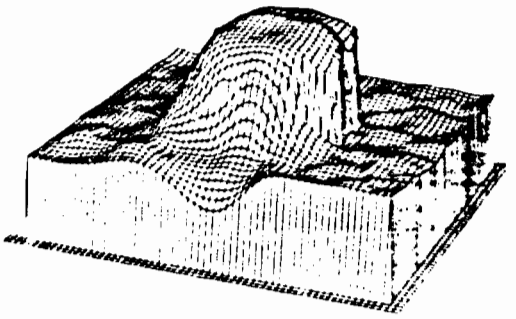


Figure 9. Final surface for the 128x128 level.

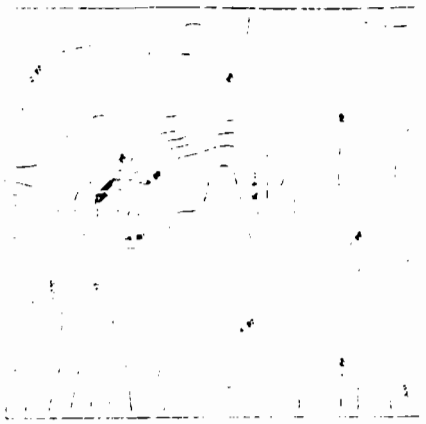


Figure 7. Contour map for the 64x64 level.

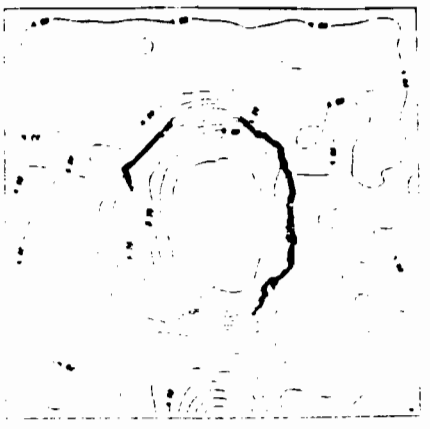


Figure 10. Contour map for the 128x128 level.

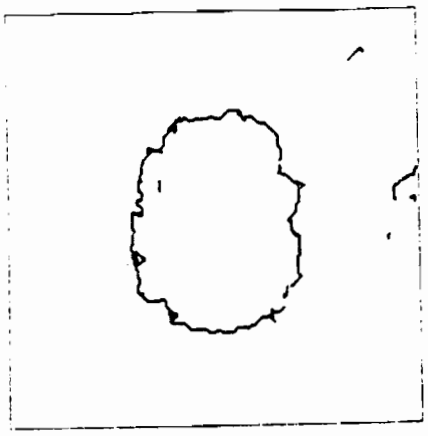


Figure 11. Depth edges detected at the 256x256 level.

Figure 14. Depth points with large (> 3) errors.

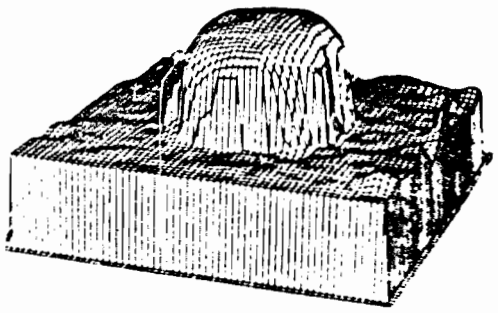


Figure 12. Final surface for the 256x256 level.

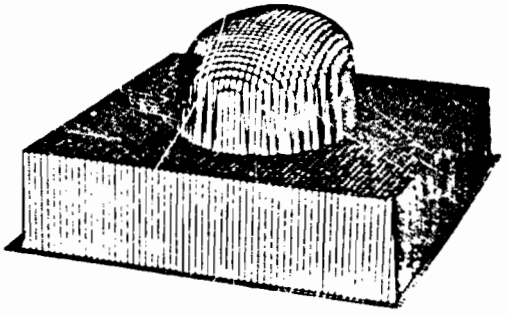


Figure 15. Surface for the ideal case (256x256).

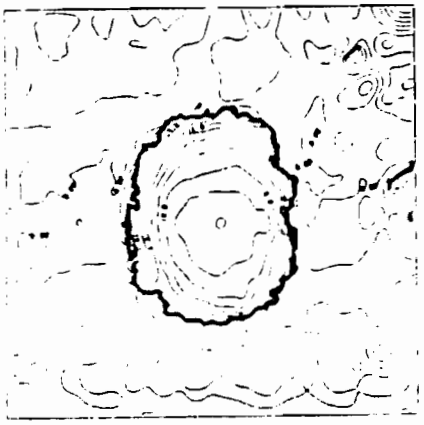


Figure 13. Contour map for the 256x256 level.

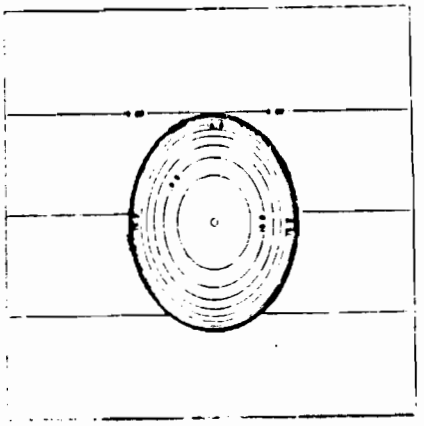


Figure 16. Contour map for the ideal case (256x256).

SECTION III

TECHNICAL REPORTS PRESENTED

GC970145-16

STRUCTURE FROM MOTION WITHOUT CORRESPONDENCE: GENERAL PRINCIPLE

Ken-ichi Kanatani*

Center for Automation Research
University of Maryland
College Park, MD 20742

ABSTRACT

A general principle is given for detecting 3D structure and motion from an image sequence without using point-to-point correspondence. The procedure consists of two stages: (i) determination of the *flow parameters*, which completely characterize the motion of the planar part of the object, and (ii) computation of 3D recovery from these flow parameters. The first stage is done by measuring *features* of the image sequence. The second stage is analytically expressed in terms of invariants with respect to coordinate changes. Typical features and relations to stepwise tracing are also discussed.

1. INTRODUCTION

Recovery of 3D structure and motion from a 2D image sequence is one of the most challenging problems in computer vision. Most existing schemes are classified into two types. One is the *correspondence-based approach*, which does not assume any particular model of the object except the rigidity of motion and uses point-to-point correspondence explicitly. The 3D structure and motion are recovered *numerically*. [1-5]. Another is the *flow-based approach*, which employs a specific model of the object and pays attention to global characteristics of the *optical flow* such as vanishing points [6-9]. This idea is fully developed by Kanatani [10-12]; if the object is a plane, the 3D structure and motion are given *analytically* in terms of invariants with respect to coordinate changes on the image plane. These invariants are derived by means of irreducible reduction of the 2D rotation group.

Although the flow-based approach does not make use of point-to-point correspondence explicitly, the optical flow itself is usually obtained by detecting the point-to-point correspondence between two successive images, and this correspondence detection is a time consuming process [13-17]. Kanatani [18-20] proposed schemes which do not use the correspondence when the object is a planar surface. In this paper, we first summarize the analytical results of Kanatani [10-12] and then generalize Kanatani's schemes [18-20] so that those analytical results can fit in the present new setting.

2. 3D MOTION FROM FLOW PARAMETERS

We assume that the image under consideration is decomposed into planar or almost planar regions, say by the method discussed by Kanatani [10,11]. Now, attention is paid to each region regarded as planar. Take a Cartesian xy-coordinate system on the image plane and the z-axis perpendicular to it. Let $z=px+qy+r$ be the equation of that plane. The coefficients p and q are the components of the *gradient* of the plane, and r represents the *absolute depth* from the image plane. Let $(0,0,r)$, the intersection between the plane and the z-axis, be a reference point (Fig. 1). The instantaneous rigid motion is specified by *translation velocity* (a,b,c) at the reference point and *rotation velocity* $(\omega_1,\omega_2,\omega_3)$ screwwise around it (i.e., with rotation axis orientation $(\omega_1,\omega_2,\omega_3)$ and angular velocity $\sqrt{\omega_1^2+\omega_2^2+\omega_3^2}$ (rad/sec) screwwise around it). Hence, our target is to reconstruct the nine *structure and motion parameters* p, q, r, a, b, c, ω_1, ω_2 and ω_3 from observation of the projected image motion.

(1) PERSPECTIVE PROJECTION

Let $(0,0,-f)$, the point on the z-axis at distance f from the image plane on the negative side, be the viewpoint or *focus* of the camera. A point (X,Y,Z) in the scene is projected to $(fX/(f+Z),fY/(f+Z))$ on the image plane. If the point is on the plane $z=px+qy+r$ which is moving as described above, it is easy to show that the following *optical flow* is induced at point (x,y) on the image plane:

$$\begin{aligned} u&=u_0+Ax+By+(Ex+Fy)x, \\ v&=v_0+Cx+Dy+(Ex+Fy)y, \end{aligned} \tag{2.1}$$

where eight *flow parameters* are given by

$$\begin{aligned} u_0&=\frac{fa}{f+r}, & v_0&=\frac{fb}{f+r}, \\ A&=p\omega_2-\frac{pa+c}{f+r}, & B&=q\omega_2-\omega_3-\frac{qa}{f+r}, \\ C&=-p\omega_1+\omega_3-\frac{pb}{f+r}, & D&=-q\omega_1+\frac{qb+c}{f+r}, \\ E&=-\frac{1}{f}(\omega_2+\frac{pc}{f+r}), & F&=\frac{1}{f}(-\omega_1+\frac{qc}{f+r}). \end{aligned} \tag{2.2}$$

In other words, what we are viewing is a very restricted form of motion whose velocities are specified only by eight flow parameters u_0, v_0, A, B, C, D, E and F. If these parameters are the same, motions seem identical to the viewer. Thus, our procedure is divided into two stages. First, we detect the flow parameters u_0, v_0, A, B, C, D, E and F from a given image sequence. Next, we compute the structure and motion parameters p, q, r, a, b, c, ω_1, ω_2 and ω_3 from these flow parameters.

* Permanent address: Department of Computer Science, Gunma University, Kiryu, Gunma 376, Japan.

The second stage is performed by solving the non-linear simultaneous equations (2.2) as follows (Appendix A): First, compute

$$U_0=u_0+iv_0, \quad T=A+D, \quad R=C-B, \quad S=(A-D)+i(B+C), \quad K=E+iF, \tag{2.3}$$

where i is the imaginary unit. Hence, U_0, K and S are complex numbers. If we put $V=a+ib$, $P=p+iq$ and $W=\omega_1+i\omega_2$, then V, c, P and ω_3 are given by

$$V=(f+r)U_0/f, \qquad c=(f+c)c',$$

$$P(c')=\frac{1}{2c'}(fK-U_0/f\pm\sqrt{(fK-U_0/f)^2-4c'S}),$$

$$W(c')=\frac{i}{2}(fK-U_0/f\mp\sqrt{(fK-U_0/f)^2-4c'S})+iU_0'/f, \tag{2.4}$$

$$\omega_3(c')=\frac{1}{2}(R+\mathrm{Re}[P(c')(W(c')^*+iU_0''/f)]),$$

$$c'=-\frac{1}{2}(T+\mathrm{Im}[P(c')(W(c')^*+iU_0''/f)]),$$

where Re[.] and Im[.] denote the real and the imaginary part respectively and * the complex conjugate. Here, P, W and ω_3 are functions of c', and c' is given by solving the last eqns (2.4). There exists only one non-zero solution c'. In fact, if we substitute the expressions for $P(c')$ and $W(c')$ in it, the equation reduces to a cubic equation in c' (Appendix A). Since an explicit form of the solution of a cubic equation exists, we can express the solution c' explicitly, although in a complicated form, if we wish. However, application of an iteration scheme seems more feasible. In any case, the problem is completely solved analytically, and we find that (i) the absolute depth r is indeterminate, (ii) $a/(f+r)$, $b/(f+r)$ and $c/(f+r)$ are uniquely determined, and (iii) there exist two sets of solutions for p, q, ω_1, ω_2 and ω_3, one being true and the other spurious. However, the spurious solution disappears if two or more planar regions of the same object are observed because ω_1, ω_2 and ω_3 must be common to them. Numerical schemes of 3D recovery from point-to-point correspondence have been known [2-4] and the existence of the spurious solution was pointed out [9], but analytical expressions like eqns (2.4) have not been known.

(2) ORTHOGRAPHIC APPROXIMATION

If we take the limit $f\to\infty$ of a large focal length f in eqns (2.2), we obtain the following *orthographic approximation*:

$$u_0=a, \qquad v_0=b,$$

$$A=p\omega_2, \quad B=q\omega_2-\omega_3, \quad C=-p\omega_1+\omega_3, \quad D=-q\omega_1, \tag{2.5}$$

$$E=0, \qquad F=0,$$

and the solution is explicitly given as follows (Appendix B):

$$V=U_0, \qquad \omega_3=\frac{1}{2}(R\pm\sqrt{SS^*-T^2}),$$

$$P=\frac{S}{k}\exp i(\frac{\pi}{4}-\frac{1}{2}\arg(S)+\frac{1}{2}\arg(2\omega_3-(R+iT))), \tag{2.6}$$

$$W=k\exp i(\frac{\pi}{4}+\frac{1}{2}\arg(S)-\frac{1}{2}\arg(2\omega_3-(R+iT))),$$

where arg denotes the argument. Here, k is an indeterminate scale factor. Thus, (i) the absolute depth r and the velocity c in the z-direction are indeterminate, (ii) an indeterminate scale factor k is involved, and (iii) there exist two types of solutions, one being true and the other spurious. However, the spurious solution disappears if two or more planar regions of the same object are observed because ω_1, ω_2 and ω_3 must be common to them. 3D recovery from point-to-point correspondence under orthographic projection was first studied by Ullman [2], and the fact that an indeterminate scale factor is necessarily involved was already pointed out [5]. However, analytical expressions of the solution have not been known.

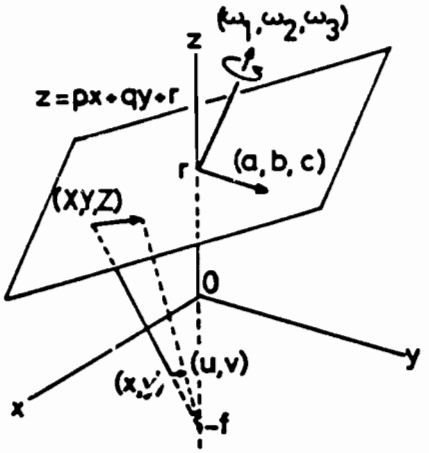


Fig. 1. A plane of equation $z=px+qy+r$ is moving with translation velocity (a,b,c) at $(0,0,r)$ and rotation velocity $(\omega_1,\omega_2,\omega_3)$ around it. An optical flow is induced on the xy-plane by perspective projection, $(0,0,-f)$ being the viewpoint.

(3) PSEUDO-ORTHOGRAPHIC APPROXIMATION

If we omit terms of $O(1/f^2)$ but retain terms of $O(1/f)$ in eqns (2.2), E and F are replaced by

$$E=\omega_2/f, \qquad F=-\omega_1/f, \tag{2.7}$$

respectively, which we call the *pseudo-orthographic approximation*. The solution is analytically given as follows (Appendix C):

$$V=(f+r)U_0/f, \qquad W=ifK, \qquad P=\frac{S}{fK-U_0/f},$$

$$\omega_3=\frac{1}{2}(R+\mathrm{Im}[Se^{-2i\alpha}]), \qquad c=-\frac{f+r}{2}(T-\mathrm{Re}[Se^{-2i\alpha}]), \tag{2.8}$$

$$\alpha=\arg(fK-U_0/f).$$

Hence, (i) the absolute depth r is indeterminate, (ii) $a/(f+r)$, $b/(f+r)$ and $c/(f+r)$ are uniquely determined, and (iii) p, q, ω_1, ω_2 and ω_3 are uniquely determined. It should be noted that *no spurious solution exists*.

The parameters of eqns (2.3) have geometrical meanings [10, 11]: U_0 *translation*, T *divergence*, R *rotation*, S *shearing* and K *fanning* (Fig. 2). They are transformed by a coordinate rotation by θ on the image plane as

$$T\to T, \qquad R\to R,$$

$$U_0\to U_0e^{-i\theta}, \qquad K\to Ke^{-i\theta}, \tag{2.9}$$

$$S\to Se^{-2i\theta}.$$

(See Appendix D.) In other words, T and R (as well as r, c and ω_3) are (absolute) invariants of *weight* 0 (or *scalars*), U_0 and K (as well as V, P and W) are (relative) invariants of weight -1 (or *vectors*), and S is a (relative) invariant of weight -2 (or a *tensor*)[12].

3. FLOW PARAMETER ESTIMATION BY FEATURES

Let $X(x,y)$ represent the image. For example, if the image consists of gray-levels, $X(x,y)$ denotes its intensity at point (x,y). If the image consists of colors, $X(x,y)$ may be a vector valued function corresponding to R, G and B. If the image consists of points and lines, $X(x,y)$ has delta-function-like singularities. In any case, we define a *feature* of image $X(x,y)$ as a *functional*, i.e., a map $F[\cdot]$ from the set of images $X(x,y)$ to real numbers.

Suppose that there is an optical flow $u(x,y)$, $v(x,y)$ on the image plane and that the image is moving according to this flow. Then, if $X(x,y)$ is an image at time t, it changes at time $t+\delta t$ after a short time interval into

$$X(x-u(x,y)\delta t, y-v(x,y)\delta t)$$

$$=X(x,y)-\frac{\partial X}{\partial x}u(x,y)\delta t-\frac{\partial X}{\partial y}v(x,y)\delta t+\cdots. \quad (3.1)$$

Then, a feature $F[X]$ at time t changes at $t+\delta t$ into $F[X]+DF[X]\delta t+\cdots$, and the *change rate* $DF[\cdot]$ is in general a *linear* functional in (x,y) and $v(x,y)$.

In view of the optical flow of eqns (2.1), this means that we have a *linear* equation of the form

$$DF[X]=C_1[X]u_0+C_2[X]v_0+\cdots+C_7[X]E+C_8[X]F, \quad (3.2)$$

where $C_1[\cdot]$, ..., $C_8[\cdot]$ are functionals derived from the given feature functional $F[\cdot]$, so that they are all known functionals. On the other hand, the change rate $DF[\cdot]$ of feature $F[\cdot]$ can be estimated by difference schemes. For example, observe the image at time t and compute feature $F(t)$. Next, observe the image at time $t+\delta t$ after a short time interval and compute the same feature $F(t+\delta t)$. Then, the time change $DF[X]$ is approximated by $(F(t+\delta t)-F(t))/\delta t$, or we can use a higher order numerical differentiation scheme if observations are made on three or more consecutive images. Thus, all quantities except u_0, v_0, A, B, C, D, E and F in eqn (3.2) are directly computed from an image sequence without requiring point-to-point correspondence. Since an equation of the form of eqn (3.2) provides a *linear constraint*, we obtain a set of simultaneous *linear* equations to solve for the flow parameters u_0, v_0, ..., E and F if we provide eight or more independent feature functionals $F_1[\cdot]$, $F_2[\cdot]$, ...

The idea of using feature functionals was suggested by Amari [21,22] and was applied to 3D recovery by Kanatani [18-20]. However, he did not divide the computation process into two stages as described here but tried to compute the structure and motion parameters p, q, r, a, b, c, ω_1, ω_2 and ω_3 directly. This leads to a set of simultaneous *non-linear* equations which are difficult to solve. He proposed an iterative scheme which traces the motion along time, starting from known initial values of p, q and r as described later. Here, however, the process is divided into two stages. We first estimate the *flow parameters* by solving a set of *linear* equations. This poses no computational problem. Then, the structure and motion parameters are computed in analytical terms as described in the previous section.

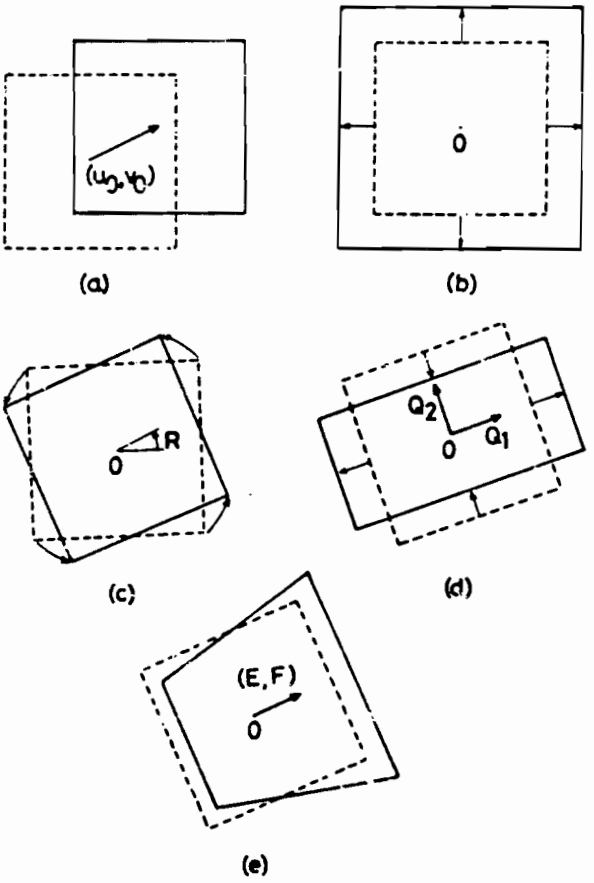


Fig. 2. (a) Translation by (u_0,v_0). (b) Divergence by T. (c) Rotation by R. (d) Shearing with $Q_1=\exp(\arg(S)/2)$ and $Q_2=iQ_1$ as axes of maximum extension and compression, respectively. (e) Fanning along (E,F).

As for the feature functionals, we can use those used by Amari [21,22] and Kanatani [18,20]. We review and modify them so that they fit in the present new setting.

(1) ANISOTROPY OF TEXTURE

Consider a surface which has a spatially nomogeneous (but not necessarily isotropic) texture consisting of line segments. The 3D structure and motion are detected by checking the *anisotropy* of the texture. This method, applicable in the case of orthographic projection, was first suggested by Witkin [23] and combined with *integral geometry* or *stereology* by Kanatani [18].

Let the line texture on the image plane be dissected into infinitesimal line elements. The orientation of each line element is specified by angle θ from the x-axis. Since there are two angles for the same orientation, i.e., θ and $\theta+\pi$ designate the same orientation, we choose one of them randomly with a probability of 1/2. Let the *distribution density* $f(\theta)$ be defined in such a way that $f(\theta)d\theta$ is the summed length of those line segments, *per unit area*, whose orientations are between θ and $\theta+d\theta$. By definition, $c_0=\int_0^{2\pi}f(\theta)d\theta$ is the total length of the

line segments per unit area. If the distribution is isotropic, $f(\theta)$ is constant for all θ. If the distribution is nearly isotropic, the distribution density $f(\theta)$ is approximated by a Fourier series up to the second order

$$f(\theta)=\frac{c_0}{2}[1+a_2\cos 2\theta+b_2\sin 2\theta],$$

$$c_0=\int_0^{2\pi} f(\theta)\,d\theta, \tag{3.3}$$

$$a_2=\frac{1}{c_0}\int_0^{2\pi} f(\theta)\cos 2\theta\,d\theta, \qquad b_2=\frac{1}{c_0}\int_0^{2\pi} f(\theta)\sin 2\theta\,d\theta.$$

Here, first order terms do not appear because of the symmetry $f(\theta+\pi)=f(\theta)$.

If the image is changing according to orthographic optical flow (i.e., eqns (2.1) with $E=G$ and $F=0$), the Fourier coefficients c_0, a_2 and b_2 of eqns (3.3) change as follows [18,29,30]:

$$D\begin{bmatrix} c_0 \\ a_2 \\ b_2 \end{bmatrix}=\frac{1}{4}\times$$

$$\begin{bmatrix} c_0(a_2-2) & c_0 b_2 & c_2 b_2 & -c_0(a_2+2) \\ -a_2^2+6 & -b_2(a_2-4) & -b_2(a_2^2+4) & a_2^2-6 \\ -a_2 b_2 & -b_2^2-4a_2+6 & -b_2^2+4a_2+6 & a_2 b_2 \end{bmatrix}\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \tag{3.4}$$

Thus, c_0, a_2 and b_2 serve as feature functionals, and eqn (3.4) corresponds to eqn (3.2), although another feature must be added to determine A, B C and D uniquely.

In order to measure c_0, a_2 and b_2 from a given image, we must estimate the distribution density $f(\theta)$ from the histogram of line segment orientations. To this end, we must choose an appropriate class interval for the histogram. If it is too large, estimation becomes crude. If it is too small, the counting for each class is greatly affected by noise. This difficulty arises because the definition of the distribution density $f(\theta)$ involves *infinitesimals*, i.e., a limit taking process.

There exists a method of estimating the distribution density $f(\theta)$ which does not involve a limit taking process. This is possible by a *stereological* technique. Instead of making a histogram, we count the number of intersections between the line segments and a probe line (or equally spaced parallel scanning lines). Let $N(\theta)$ be the number of intersections per unit length of the scanning line of orientation θ. Then, the observed intersection count $N(\theta)$ is related to the distribution density $f(\theta)$ by what Kanatani [18, 30] called the (two-dimensional) *Buffon transform*:

$$N(\theta)=\int_0^{2\pi}|\sin(\theta-\theta')|f(\theta')\,d\theta'. \tag{3.5}$$

If the distribution density $f(\theta)$ is given by eqns (3.3), the intersection count $N(\theta)$ becomes [18, 30]

$$N(\theta)=\frac{C_0}{2}[1+A_2\cos 2\theta+B_2\sin 2\theta],$$

$$C_0=\int_0^{2\pi} f(\theta)\,d\theta, \tag{3.6}$$

$$A_2=\frac{1}{C_0}\int_0^{2\pi} N(\theta)\cos 2\theta\,d\theta, \qquad B_2=\frac{1}{c_0}\int_0^{2\pi} N(\theta)\sin 2\theta\,d\theta.$$

where

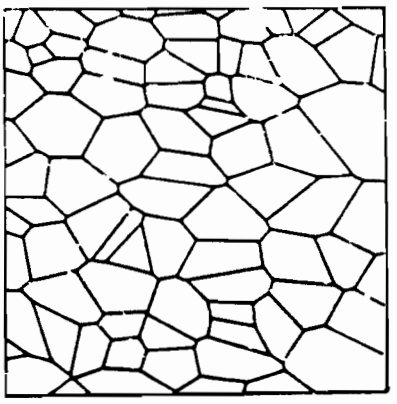


Fig. 3. An example of a textured surface image.

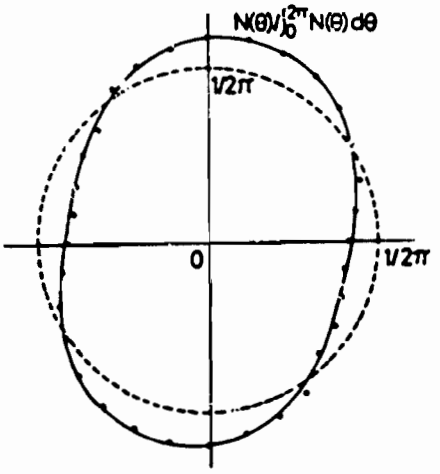


Fig. 4. The number of intersections of the texture of Fig. 1 with parallel scanning lines of different orientation, the spacing being 1/22 of the side of the square frame. The data are normalized so that the average is $1/2\pi$. The solid curve is Fourier approximation up to the second order.

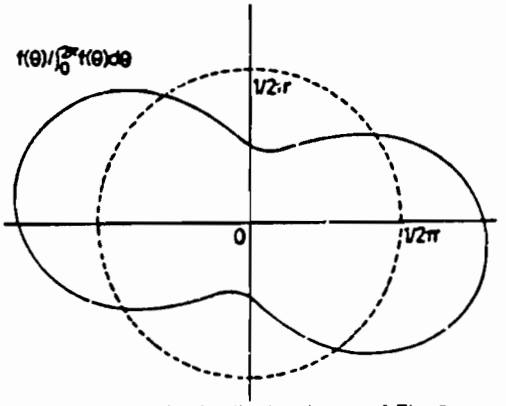


Fig. 5. Estimation of the distribution density of Fig. 3 up to the second order Fourier harmonics.

$$C_0=4c_0, \quad A_2=-\frac{1}{3}a_2, \quad B_2=-\frac{1}{3}b_2. \tag{3.7}$$

Hence, we can use C_0, A_2 and B_2 themselves as feature functionals. They are computed by measuring the intersection count $N(\theta)$ and approximating the integrations of eqns (3.6) by appropriate summations. For example, putting $N_k=N(\pi k/N)$, $k=0,1,\ldots,N-1$, we may adopt the approximation

$$C=2\sum_{k=0}^{N-1}N_k/N,$$
$$A_2=2\sum_{k=0}^{N-1}N_k\cos\frac{2\pi k}{N}/\sum_{k=0}^{N-1}N_k, \quad B_2=2\sum_{k=0}^{N-1}N_k\sin\frac{2\pi k}{N}/\sum_{k=0}^{N-1}N_k. \tag{3.8}$$

Consider Fig. 3, for example. If we draw on it equally spaced parallel scanning lines whose spacing is 1/22 of one side of the square frame for orientations $\theta_k=\pi k/16$, $k=0,1,\ldots,15$ with $N=16$, i.e., at 11.25° intervals, we obtain the intersection count as shown in Fig. 4, from which we obtain $A_2=-0.172$ and $B_2=0.068$. The solid curve is the corresponding approximation of eqns (3.6). Fig. 5 is the recovered distribution density of eqns (3.3) estimated by using eqns (3.7).

From eqns (3.6) and (3.10), the change rates of C_0, A_2 and B_2 become as follows:

$$\mathrm{D}\begin{bmatrix}C_0\\A_2\\B_2\end{bmatrix}=\frac{3}{4}\times$$
$$\begin{bmatrix}-C_0(A_2+\frac{2}{3}) & -C_0B_2 & C_0B_2 & C_0(A_2-\frac{2}{3})\\ A_2^2-\frac{2}{3} & B_2(A_2+\frac{4}{3}) & B_2(A_2-\frac{4}{3}) & -A_2^2+\frac{2}{3}\\ A_2B_2 & B_2^2-\frac{4}{3}A_2-\frac{2}{3} & B_2^2-\frac{4}{3}A_2-\frac{2}{3} & -A_2B_2\end{bmatrix}\begin{bmatrix}A\\B\\C\\D\end{bmatrix}. \tag{3.9}$$

(2) ANISOTROPY OF CONTOUR

In the above, we assumed *spatial homogeneity*, since anisotropy is expressed *per unit area*. This assumption assures that the portion of the texture newly coming into view has the same statistical characteristics as the portion of the texture going out of view. However, this assumption is not necessary if the entire planar region is viewed, i.e., if we can always identify the planar region that we are looking at. Then, the distribution density $f(\theta)$ is defined in such a way that $f(\theta)d\theta$ is just the summed length (not per unit area) of those line segments whose orientations are between θ and $\theta+d\theta$. By definition, $c_0=\int_0^{2\pi}f(\theta)d\theta$ is the total length of the line segments. If the distribution is isotropic, $f(\theta)$ is constant for all θ. If the distribution density $f(\theta)$ is approximated by the Fourier series (3.3) up to the second order, the change rates of c_0, a_2 and b_2 are given by eqn (3.4) except that the first row of the matrix is replaced by

$$c_0(a_2+2) \quad c_0b_2 \quad c_0b_2 \quad -c_0(a_2-2). \tag{3.10}$$

If we count the number of intersections between the texture of the entire planar region in question and a probe line (or equally spaced parallel scanning lines), and if $N(\theta)$ is the number of intersections per unit length of the scanning line of orientation θ, then $N(\theta)$ and $f(\theta)$ are again related by the Buffon transform of eqn (3.5). Hence, if the distribution den-

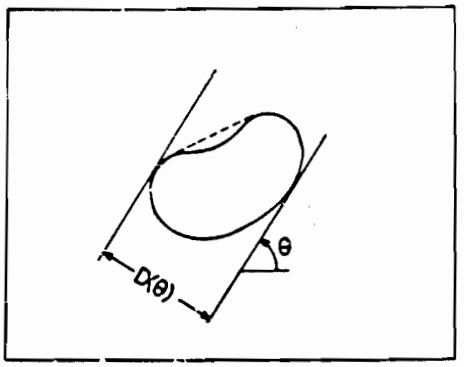


Fig. 6. The caliper diameter $D(\theta)$ is the distance between two parallel lines tangent to the contour from outside.

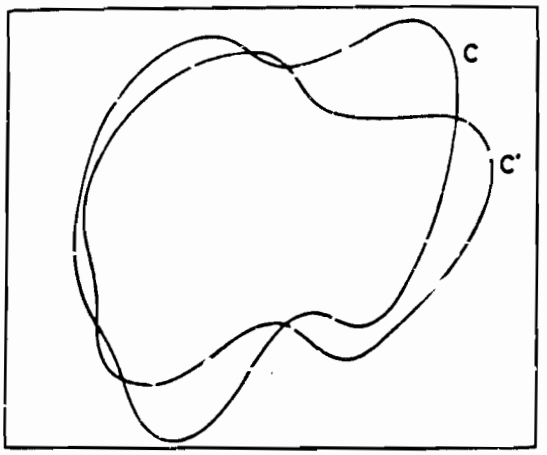


Fig. 7. Two contour images C and C' of the same planar surface.

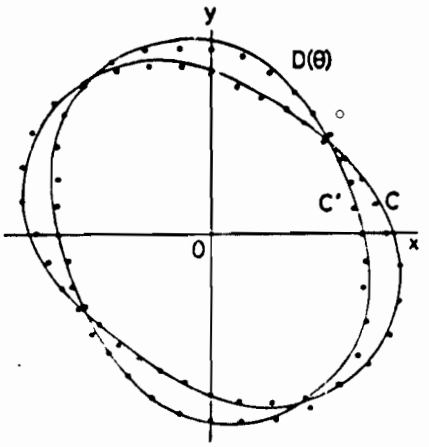


Fig. 8. Diameters of the contours C and C' of Fig. 7 for different orientations - white circles for C and black dots for C'. The solid curves are Fourier approximation up to the second order.

sity is approximated by eqn (3.3), $N(\theta)$ is given by the form of eqn (3.6), and the change rates of C_0, A_2 and B_2 are given by eqn (3.9) except that the first row of the matrix is replaced by

$$-C_0(A_2-\frac{2}{3}) \quad -C_0B_2 \quad 2C_0(A_2+\frac{2}{3}). \tag{3.11}$$

An interesting application arises when the planar region has no texture but its contour is viewed. Then, the contour itself can be regarded as a texture. If the contour shape is convex, the intersection counting is equivalent to measuring the *diameter* $D(\theta)$ defined as the spacing of two parallel lines of orientation θ tangent to the contour (Fig. 6), for every line has two intersections if they exist (excluding the exceptional case of tangency). The contour shape need not be convex if the diameter is measured from outside, for in this case the *convex hull* of the contour plays the role of a texture. The convex hull is *invariant* with respect to projection; the convex hull of a projected contour is the same as the projection of the convex hull of the original contour. The diameter $D(\theta)$ and the distribution density $f(\theta)$ of the contour are related as follows [19]:

$$D(\theta)=\frac{1}{2}\int_0^{2\pi}|\sin(\theta-\theta')|f(\theta')d\theta'. \tag{3.12}$$

If this function is expressed in Fourier series as in eqn (3.6), the coefficients C_0, A_2 and B_2 change as in eqn (3.9) with the first row replaced by (3.11). Consider the two contour images C and C' of Fig. 7, for example. The diameters measured at 10° intervals of orientation are plotted in Fig. 8, where the white circles correspond to C and the black ones to C'. The solid curves are approximations of the form of eqn (3.6) with C_0, A_2 and B_2 computed by eqns (3.8), indicating that they fairly well characterize the data.

(3) FILTERING GRAY-LEVEL IMAGES

Suppose we are observing a sequence of gray-level images of a planar region. Amari [21, 22] suggested the use of *filtering* or *weighted averaging* for feature detection. Namely, we use

$$F[X]=\iint_W m(x,y)X(x,y)dxdy, \tag{3.13}$$

as a feature, where $m(x,y)$ is a fixed *weight function* of the filter, and integration is done over a fixed domain or *window* W on the image plane. Suppose the area of non-zero gray-levels is localized in the window W so that $X(x,y)=0$ along the window boundary and suppose the gray-level does not depend on the gradient or the depth of the object surface. An example is letters, lying entirely in the window W, drawn on a white (or black) object surface.

If the image $X(x,y)$ changes according to eqn (3.1), the feature $F[X]$ becomes after a short time interval δt

$$\iint_W m(x,y)dxdy-\iint_W m(x,y)(\frac{\partial X}{\partial x}u(x,y)+\frac{\partial X}{\partial y}v(x,y))\delta t dxdy+\cdots$$

$$=F[X]+\iint_W(\frac{\partial um}{\partial x}+\frac{\partial vm}{\partial y})X\delta t dxdy+\cdots, \tag{3.14}$$

where we performed integration by parts, setting integrals along the window boundary to be zero according to our assumption that $X(x,y)$ is zero at the window boundary. Thus, the change rate $DF[X]$ of the feature $F[X]$ is given by

$$DF[X]=\iint_W(\frac{\partial u}{\partial x}m+\frac{\partial v}{\partial y}m+u\frac{\partial m}{\partial x}+v\frac{\partial m}{\partial y})Xdxdy. \tag{3.15}$$

When the optical flow is given by eqns (2.1), functionals $C_1[\cdot]$, ..., $C_8[\cdot]$ of eqn (3.2) become as follows:

$$C_1[X]=\iint_W m_xXdxdy, \quad C_2[X]=\iint_W m_yXdxdy,$$

$$C_3[X]=\iint_W(m+xm_x)Xdxdy, \quad C_4[X]=\iint_W ym_xXdxdy,$$

$$C_5[X]=\iint_W xm_yXdxdy, \quad C_6[X]=\iint_W(m+ym_y)Xdxdy, \tag{3.16}$$

$$C_7[X]=\iint_W(3xm+x^2m_x+xym_y)Xdxdy,$$

$$C_8[X]=\iint_W(3ym+xym_x+y^2m_y)Xdxdy,$$

where $m_x=\partial m/\partial x$ and $m_y=\partial m/\partial y$ are known functions. Thus, $C_1[\cdot]$, ..., $C_8[\cdot]$ can be implemented as filters. Here, we assumed that $X(x,y)=0$ at the window boundary. This assumption is not essential, and it can be removed. Instead the expressions of the functionals $C_1[\cdot]$, ..., $C_8[\cdot]$ include terms of line integral along the window boundary.

(4) INTEGRATION ALONG AND INSIDE THE CONTOUR

Kanatani [20] considered the case where only the bounding contour of a planar region is observed. He proposed the use of integration along the contour C of a given fixed function $m(x,y)$,

$$F[X]=\int_C m(x,y)ds, \tag{3.17}$$

as a feature, where ds denotes the line element along the contour C. This integration is easily performed on the image by using a scheme of numerical integration [20]. Then, we see that

$$DF[X]=$$

$$\int_C[u\frac{\partial m}{\partial x}+v\frac{\partial m}{\partial y}+(\frac{\partial u}{\partial x}x'^2+(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x})x'y'+\frac{\partial v}{\partial y}y'^2)m]ds, \tag{3.18}$$

where $x'=dx/ds$ and $y'=dy/ds$. When the optical flow is given by eqns (2.1), functionals $C_1[\cdot]$, ..., $C_8[\cdot]$ of eqn (3.2) become as follows:

$$C_1[X]=\int_C m_xds, \quad C_2[X]=\int_C m_yds,$$

$$C_3[X]=\int_C[xm_x+x'^2m]ds, \quad C_4[X]=\int_C[ym_x+x'y'm]ds,$$

$$C_5[X]=\int_C[xm_y+x'y'm]ds, \quad C_6[X]=\int_C[ym_y+y'^2m]ds, \tag{3.19}$$

$$C_7[X]=\int_C[x^2m_x+xym_y+(2xx'^2+yx'y'+xy'^2)m]ds,$$

$$C_8[X]=\int_C[xym_x+y^2m_y+(yx'^2+xx'y'+2yy'^2)m]ds.$$

Hence, $C_1[\cdot]$, ..., $C_8[\cdot]$ can be computed on the image plane by using a scheme of numerical integration.

Kanatani [9] also proposed the use of surface integration inside the planar region S

$$F[X]=\iint_S m(x,y)dxdy, \tag{3.20}$$

of a fixed function $m(x,y)$. Now, integration is done over a moving region S, not over a fixed window W. The change rate is expressed in two ways, due to *Green's theorem*, as follows:

$$DF[X]=\iint_S[u\frac{\partial m}{\partial x}+v\frac{\partial m}{\partial y}+(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y})m]dxdy$$

$$=\int_C(uy'-vx')mds. \tag{3.21}$$

When the optical flow is given by eqns (2.1), functionals $C_1[\cdot]$, ..., $C_8[\cdot]$ of eqn (3.2) become as follows:

$$C_1[X]=\int_C my'ds=\iint_S m_x dxdy, \quad C_2[X]=-\int_C mx\,ds=\iint_S m_y dxdy,$$

$$C_3[X]=\int_C xy'mds=\iint_S\{m+xm_x\}dxdy,$$

$$C_4[X]=\int_C yy'mds=\iint_S ym_x dxdy,$$

$$C_5[X]=-\int_C xx'mds=\iint_S xm_y dxdy, \tag{3.22}$$

$$C_6[X]=-\int_C yx'mds=\iint_S\{m+ym_y\}dxdy,$$

$$C_7[X]=\int_C(x^2y'-xyx')mds=\iint_S\{3xm+x^2m_x+xym_y\}dxdy,$$

$$C_8[X]=\int_C(xyy'-y^2x')mds=\iint_S\{3ym+xym_x+y^2m_y\}dxdy.$$

Hence, $C_1[\cdot]$, ..., $C_8[\cdot]$ are computed on the image plane as either line integrals or surface integrals.

4. STEPWISE TRACING AND STEREO

According to the method described so far, the flow parameters u_0, v_0, A, B, C, D, E and F can be extracted from two (or more) consecutive images, and then the structure and motion parameters a, b, c, p, q, ω_1, ω_2 and ω_3 are determined by analytical equations. As was shown, however, there remain certain indeterminacies including the absolute depth r. These indeterminacies can be removed if a sequence of images is available and if the initial position of the surface is known [19, 20]. This becomes possible if we note the fact that if a plane $z=px+qy+r$ is moving with translation velocities a, b and c and rotation velocities ω_1, ω_2 and ω_3 as described in Section 2, the coefficients p, q and r change as

$$\frac{dp}{dt}=pq\omega_1-(p^2+1)\omega_2-q\omega_3, \quad \frac{dq}{dt}=(q^2+1)\omega_1-pq\omega_2+p\omega_3,$$
$$\frac{dr}{dt}=c-pa-qb. \tag{4.1}$$

Suppose p, q and r are known at time t. Substitution of eqns (2.2) in eqn (3.2) yields

$$DF[X]=C_a[X]+C_b[X]+C_c[X]+C_{\omega_1}[X]+C_{\omega_2}[X]+C_{\omega_3}[X], \tag{4.2}$$

where $C_a[\cdot]$, $C_b[\cdot]$, $C_c[\cdot]$, $C_{\omega_1}[\cdot]$, $C_{\omega_2}[\cdot]$ and $C_{\omega_3}[\cdot]$ are functionals defined by

$$C_a[\cdot]=\frac{1}{f+r}(fC_1[\cdot]-pC_3[\cdot]-qC_4[\cdot]), \quad C_b[\cdot]=\frac{1}{f+r}(fC_2[\cdot]-pC_5[\cdot]-qC_6[\cdot]),$$

$$C_c[\cdot]=-\frac{1}{f+r}(C_3[\cdot]+C_6[\cdot])-\frac{1}{f}(pC_7[\cdot]+qC_8[\cdot]),$$

$$C_{\omega_1}[\cdot]=-(pC_5[\cdot]+qC_6[\cdot])+\frac{1}{f}C_3[\cdot]), \quad C_{\omega_2}[\cdot]=pC_3[\cdot]+qC_5[\cdot]+\frac{1}{f}C_7[\cdot], \tag{4.3}$$

$$C_{\omega_3}[\cdot]=C_5[\cdot]-C_4[\cdot].$$

Since p, q and r are known, $C_a[\cdot]$, ..., $C_{\omega_3}[\cdot]$ are known functionals. The left-hand side of eqn (4.2), i.e., the change rate $F[X]$ of feature $F[X]$, is obtained by a numerical differentiation scheme as described earlier. Hence, if we use six or more independent feature functionals, we obtain a set of simultaneous *linear* equations of the form of eqn (4.2) to determine a, b, c, ω_1, ω_2 and ω_3. Then p, q and r at time $t+\delta t$ are determined by integrating eqns (4.1) by some numerical integration scheme like

$$p\leftarrow p+\{pq\omega_1-(p^2+1)\omega_2-\omega_3\}\delta t, \quad q\leftarrow q+\{(q^2+1)\omega_1-pq\omega_2+p\omega_3\}\delta t,$$
$$r\leftarrow r+\{c-pa-qb\}\delta t. \tag{4.4}$$

or some other higher order scheme. This process is repeated to determine the course of motion uniquely along time [19, 20]. During this process, small errors at each step may accumulate, so that appropriate modifications are necessary once in a while, say, by the direct method described earlier or some other source of information.

This method is also used to determine the surface orientation and position p, q and r from *stereo vision* without using point-to-point correspondence. If we move the camera by l in the negative x-direction, the object moves by l in the x-direction relative to the camera. In view of eqn (4.2), the change rate $dF[X]/dl$ of feature $F[X]$ is equal to $C_a[X]$. Similarly, $C_b[X]$ and $C_c[X]$ are directly obtained by moving the camera in the y- and the z-direction and measuring the change rate of feature $F[X]$. (In practice, of course, the camera need not be moved if the necessary number of cameras are appropriately positioned beforehand.) Then, the first three of eqns (4.3) provide a set of simultaneous equations to solve for p, q and r, since $C_a[X]$, ..., $C_{\omega_3}[X]$ are also measured on the image. First, p and q are given as a solution of

$$\begin{bmatrix} C_c[X]C_3[X]+\frac{1}{f}C_a[X]C_7[X] & C_c[X]C_4[X]+\frac{1}{f}C_a[X]C_8[X] \\ C_c[X]C_5[X]+\frac{1}{f}C_b[X]C_7[X] & C_c[X]C_6[X]+\frac{1}{f}C_b[X]C_8[X] \end{bmatrix}\begin{bmatrix} p \\ q \end{bmatrix}$$
$$=\begin{bmatrix} fC_c[X]C_1[X]+C_a[X](C_3[X]+C_6[X]) \\ fC_c[X]C_2[X]+C_b[X](C_3[X]+C_6[X]) \end{bmatrix}, \tag{4.5}$$

and r is given by

$$r=\frac{fC_1[X]-pC_3[X]-qC_4[X]}{C_a[X]}f=\frac{fC_2[X]-qC_5[X]-qC_6[X]}{C_b[X]}f. \tag{4.6}$$

(If we use more than three independent feature functionals, the camera need be moved in only one direction, say, in the x-direction alone. However, this does not seem to be feasible in view of noise susceptibility.)

In the orthographic approximation $f\to\infty$, eqns (4.3) become

$$C_a[\cdot]=C_1[\cdot], \quad C_b[\cdot]=C_2[\cdot], \quad C_c[\cdot]=0,$$

$$C_{\omega_1}[\cdot]=-(pC_5[\cdot]+qC_6[\cdot]), \quad C_{\omega_2}[\cdot]=pC_3[\cdot]+qC_5[\cdot], \tag{4.7}$$

$$C_{\omega_3}[\cdot]=C_5[\cdot]-C_4[\cdot],$$

and the process goes similarly except that c is not determined, as is obvious for orthographic projection. If the feature functionals that we use are invariant with respect to translations as in (1) and (2) of the previous section, only three such features are necessary to compute ω_1, ω_2 and ω_3, which in turn determine the *trajectory* of p and q. Fig. 10 shows the trajectory of the motion of Fig. 9 obtained by measuring the diameter $D(\theta)$ [19]. However, special care should be taken when $p=0$ and $q=0$, in which case both $C_{\omega_1}[X]$ and $C_{\omega_2}[X]$ vanish and hence ω_1 and ω_2 are not determined. In this case, we must use a higher order expression of the optical flow as shown in [18, 19].

In the pseudo-orthographic approximation, the process goes similarly except that $C_c[.]$ of eqns (4.3) is replaced by

$$C_c[.]=-\frac{1}{f+r}(C_3[.]+C_6[.]). \tag{4.8}$$

Acknowledgement. The author wants to express his special thanks to Professor Azriel Rosenfeld and Professor Larry S. Davis at the University of Maryland for helpful comments. He also wants to thank Professor Shun-ichi Amari at Tokyo University, Dr. Allen Waxman at Thinking Machines Corporation and Mr. Muralidnara Subbarao at the University of Maryland for discussions and suggestions.

REFERENCES

[1] S. Ullman, The interpretation of structure from motion, *Proc. R. Soc. Lond.*, **B-203** (1979), 405 - 426.

[2] H.-H. Nagel, Representation of moving rigid objects based on visual observations, *Computer*, **14-8** (1981), 29 - 39.

[3] H. C. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, *Nature*, **236** (1981), 133 - 135.

[4] R. Y. Tsai and T. S. Huang, Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces, *IEEE Trans. Pattern Anal. Machine Intell.*, **PAMI-6** (1984), 13 - 27.

[5] K. Sugihara and N. Sugie, Recovery of rigid structure from orthographically projected optical flow, *Comput. Vision Graphics Image Processing*, **27** (1984), 309 - 320.

[6] D. A. Gordon, Static and dynamic visual fields in human space perception, *J. Opt. Soc. Am.*, **55** (1965), 1296 - 1303.

[7] W. F. Clocksin, Perception of surface slant and edge labels from optical flow: A computational approach, *Perception*, **9** (1980), 253 - 269.

[8] J. J. Koenderink and A. J. van Doorn, Exterospecific component of the motion parallax field, *J. Opt. Soc. Am.*, **71** (1981), 953 - 957.

[9] H. C. Longuet-Higgins, The visual ambiguity of a moving plane, *Proc. R. Soc. Lond.*, **B-223** (1984), 165 - 175.

[10] K. Kanatani, *Analysis of Structure and Motion from Optical Flow: Part I Orthographic Projection*, Technical Report, University of Maryland, 1985.

[11] K. Kanatani, *Analysis of Structure and Motion from Optical Flow: Part II Central Projection*, Technical Report, University of Maryland, 1985.

[12] K. Kanatani, *Analysis of Structure and Motion from Optical Flow: Part III Invariant Decomposition*, Technical Report, University of Maryland, 1985.

[13] S. Ullman, *The Interpretation of Visual Motion*, MIT Press, Cambridge, Mass., 1979.

[14] R. Jain, Dynamic scene analysis using pixel-based processes, *Computer*, **14-8** (1981), 12 - 18.

[15] W. B. Thompson, Lower-level estimation and interpretation of visual motion, *Computer*, **14-8** (1981), 20 - 28.

[16] B. K. P. Horn and B. G. Schunk, Determination of optical flow, *Artif. Intell.*, **17** (1981), 185 - 203.

[17] J. M. Prager and M. A. Arbib, Computing optic flow: the MATCH algorithm and prediction, *Comput. Vision Graphics Image Processing*, **24** (1983), 271 - 304.

[18] K. Kanatani, Detection of surface orientation and motion from texture by a stereological technique, *Artif. Intell.*, **23** (1984), 213 - 237.

[19] K. Kanatani, Tracing planar surface motion from projection without knowing the correspondence, *Comput. Vision Graphics Image Processing*, **29** (1985), 1 - 12.

[20] K. Kanatani, Detecting the motion of a planar surface by line and surface integrals, *Comput Vision Graphics Image Processing*, **29** (1985), 13 - 22.

[21] S. Amari, Invariant structures of signal and feature spaces in

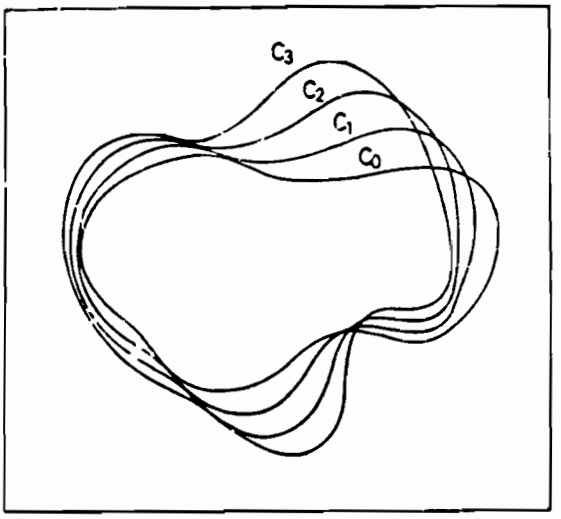


Fig. 9. Contours of a moving plane viewed orthographically. The orientation of C_0 is assumed to be known.

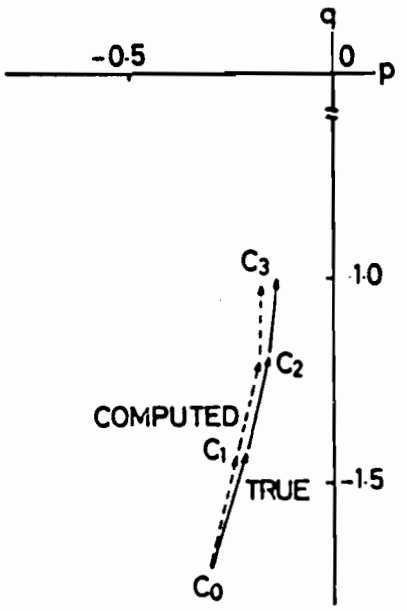


Fig. 10. The true and the computed trajectory of the gradient (p,q) obtained by measuring the diameter $D(\theta)$ of the contours of Fig. 9 at 10° intervals.

pattern recognition problems, *RAAG Memoirs*, **4** (1968), 553 - 566

[2] S. Amari, Feature spaces which admit and detect invariant signal transformations, *Proc. 4th Int. Conf. Pattern Recog.*, 1978, pp. 452 - 456.

[3] A. P. Witkin, Recovering surface shape and orientation from texture, *Artif. Intell.*, **17** (1981), 17 - 45.

[4] M. G. Kendall and P. A. Moran, *Geometrical Probability*, Charles Griffin, London, 1963.

[5] R. T. DeHoff and F. N. Rhines, *Quantitative Microscopy*. McGraw-Hill, New York, 1968.

[6] E. E Underwood, *Quantitative Stereology*, Addison-Wesley, Reading, Mass., 1970.

[7] L. A. Santalo, *Integral Geometry and Geometric Probability*, Addison-Wesley, Reading, Mass., 1976.

[8] E. R. Weibel, *Stereological Methods*, Vols. 1, 2, Academic Press, New York, 1979, 1980.

[9] K. Kanatani, Distribution of directional data and fabric tensors, *Int. J. Engng Sci.*, **22** (1984), 149 - 164.

[10] K. Kanatani, Stereological determination of structural anisotropy, *Int. J. Engng Sci.*, **22** (1984), 531 - 546.

APPENDIX A

If we substitute eqns (2.2) in eqns (2.3), we obtain

$$U_0=\frac{f(a+ib)}{f+r},$$

$$T=p\omega_2-q\omega_1-\frac{pa+qb+2c}{f+r}, \quad R=-p\omega_1-q\omega_2+2\omega_3-\frac{pb-qc}{f+r},$$

$$S=p\omega_2+q\omega_1-\frac{pa-qb}{f+r}+i(q\omega_2-p\omega_1-\frac{pb+qa}{f+r}), \tag{A.1}$$

$$K=\frac{1}{f}\omega_2+\frac{cp}{f(f+r)}+i(-\frac{1}{f}\omega_1+\frac{cq}{f(f+r)}).$$

If we put $V=a+ib$, $P=p+iq$ and $W=\omega_1+i\omega_2$, these equations are rewritten as

$$U_0=\frac{fV}{f+r},$$

$$R+iT=2\omega_3-\frac{2ic}{f+r}-P(W^*+\frac{i}{f}U_0^*). \tag{A.2}$$

$$S=-iP(W-\frac{i}{f}U_0), \quad K=-\frac{i}{f}W+\frac{cP}{f(f+r)}.$$

Putting

$$c'=\frac{c}{f+r}, \quad W'=iW-\frac{i}{f}U_0, \tag{A.3}$$

the above equations are further rewritten as

$$V=\frac{f+r}{f}U_0, \tag{A.4}$$

$$PW'^*=(2\omega_3-R)-i(2c'+T), \tag{A.5}$$

$$PW'=iS, \quad c'P-iW'=fK-\frac{1}{f}U_0. \tag{A.6}$$

Since V is given by eqn (A.4), the remaining equations are the equations to determine c', P, W' and ω_3.

First, we check whether $c'=0$ or not. If so, we have $W'=i(fK-U_0/f)$ from the second of eqns (A.6). Then, $P=S/(fK-U_0/f)$ from the first. We can conclude $c'=0$ if and only if these W' and P satisfy $PW'^*=(2\omega_3-R)-iT$ obtained from eqn (A.5). If this is satisfied (within a certain threshold), ω_3 is given by $\omega_3=(R+\mathrm{Re}[PW'^*])/2$.

Suppose we have already checked that c' is not zero. The first of eqns (A.6) is rewritten as $(c'P)(-iW')=c'S$. Hence, eqns (A.6) means that $c'P$ and $-iW'$ are the two roots of the quadratic equation

$$X^2-LX+c'S=0 \qquad (L\equiv fK-U_0/f). \tag{A.7}$$

Hence, P and W' are given as functions of c' by

$$P(c')=\frac{1}{2c'}(L\pm\sqrt{L^2-4c'S}), \quad W'(c')=\frac{i}{2}(L\pm\sqrt{L^2-4c'S}). \tag{A.8}$$

Then, eqn (A.5) gives ω_3 as a function of c' by

$$\omega_3=\frac{1}{2}(R+\mathrm{Re}[P(c')W'(c')^*]), \tag{A.9}$$

and the equation to determine c' is

$$c'=-\frac{1}{2}(T+\mathrm{Im}[P(c')W'(c')^*]). \tag{A.10}$$

Eqn (A.10) defines a unique equation although two sets of solutions exist for P, W' and ω_3. To see this, let X_1 and X_2 be the two roots of eqn (A.7). If we choose $P=X_1/c'$ and $W'=iX_2$, we have $\mathrm{Im}[PW'^*]=-\mathrm{Re}[X_1X_2^*]/c'$, while if we choose $P=X_2/c'$ and $W'=iX_1$, we have $\mathrm{Im}[PW'^*]=-\mathrm{Re}[X_1^*X_2]/c'$. Since $\mathrm{Re}[X_1X_2^*]=\mathrm{Re}[X_1^*X_2]$, $\mathrm{Im}[PW'^*]$ of eqn (A.10) remains the same for both cases.

If we actually substitute eqns (A.8) in eqn (A.10), we obtain

$$\sqrt{16|S|^2c'^2-8\mathrm{Re}[L^2S^*]c'+|L|^4}=-8c'^2-4Tc'+|L|^2. \tag{A.11}$$

The left-hand side is a smooth concave function (or a constant if $S=0$) passing through $(0,|L|^2)$, while the right-hand side is a smooth convex quadratic function also passing through $(0,|L|^2)$ (Fig. A). Since we know that $c'\neq0$, there exists a single unique non-zero solution c'.

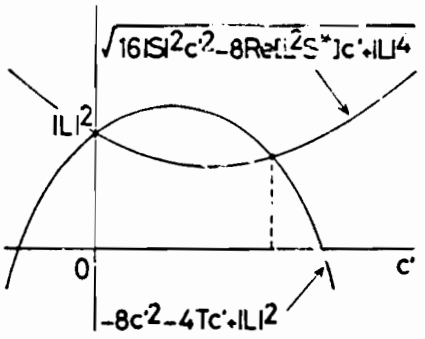


Fig. A. Existence and uniqueness of nonzero c'.

If we take the squares of both sides, we obtain a cubic equation

$$c'^3+Tc'^2+\frac{1}{4}(T^2-|S|^2-|L|^2)c'^2+\frac{1}{8}(T|S|^2-2\mathrm{Re}[S^*L^2]-T|L|^2)=0 \tag{A.12}$$

From Fig. A, it is easy to see that this cubic equation has three real roots and that the middle one is the desired root.

(The other two roots were introduced by squaring of both sides.)

APPENDIX B

Since $u_0=a$ and $v_0=b$, we only need to determine p, q, ω_1, ω_2 and ω_3. (c and r are indeterminate due to orthography.) If we substitute eqns (2.5) in eqns (2.3), we obtain

$$T=p\omega_2-q\omega_1, \quad R=2\omega_3-p\omega_1-q\omega_2,$$
$$S=p\omega_2-q\omega_1+i(q\omega_2-p\omega_1). \tag{B.1}$$

The first two equations are combined into a single equation

$$R+iT=2\omega_3-p\omega_1-q\omega_2+i(p\omega_2-q\omega_1). \tag{B.2}$$

If we put $P=p+iq$ and $W=\omega_1+i\omega_2$, the equations become

$$PW^*=2\omega_3-(R+iT), \quad PW=iS. \tag{B.3}$$

Since $|PW^*|=|PW|$, the right-hand sides must have the same modulus, i.e.,

$$(2\omega_3-(R+iT))(2\omega_3-(R-iT))=SS^*, \tag{B.4}$$

from which ω_3 is given by

$$\omega_3=\frac{1}{2}(R\pm\sqrt{SS^*-T^2}). \tag{B.5}$$

From eqns (B.3), we immediately see that if W and P are a solution, then so are kW and P/k where k is an arbitrary non-zero real constant. Hence, we do not lose generality if we put $W=k\exp(i\arg(W))$, where k is an indeterminate scale factor. Eliminating P from eqns (B.3) by taking ratios of both sides, we obtain

$$\frac{W}{W^*}=\frac{iS}{2\omega_3-(R+iT)}. \tag{B.6}$$

Taking the argument of both sides yields

$$2\arg(W)=\frac{\pi}{2}+\arg(S)-\arg(2\omega_3-(R+iT)) \pmod{2\pi}. \tag{B.7}$$

and hence

$$\arg(W)=\frac{\pi}{4}+\frac{1}{2}\arg(S)-\frac{1}{2}\arg(2\omega_3-(R+iT)) \pmod{\pi}. \tag{B.8}$$

However, we can ignore the mod π by allowing the scale factor k to be negative. Then, W is given by the second of eqns (2.6). Finally, P is given from the second of eqns (B.3) by $P=iS/W$, and hence it is written as in eqns (2.6).

APPENDIX C

If the pseudo-orthographic approximation (2.7) is adopted, eqns (A.6) are replaced by

$$PW'=iS, \quad W=ifK \tag{C.1}$$

Hence, W is explicitly obtained, and $P=iS/W=S/(fK-U_0/f)$. The remaining ω_3 and c are given from eqn (A.5) as

$$\omega_3=\frac{1}{2}(R+\mathrm{Re}[PW'^*]), \quad c=\frac{f+r}{2}(T+\mathrm{Im}[PW'^*]) \tag{C.2}$$

If we note that

$$PW'^*=-iS\frac{fK-U_0/f}{fK-U_0/f}=-iSe^{-2i\alpha}, \tag{C.3}$$

we obtain eqns (2.8).

APPENDIX D

Optical flows are observed in the form of eqns (2.1) with respect to an xy-coordinate system arbitrarily fixed on the image plane. The choice of the coordinate system is completely arbitrary. Suppose we use an $x'y'$-coordinate system obtained by rotating the xy-coordinate system by angle θ counterclockwise. Then, the optical flow must bear the same form

$$u'=u_0'+A'x'+B'y'+(E'x'+F'y')x',$$
$$v'=v_0'+C'x'+D'y'+(E'x'+F'y')y', \tag{D.1}$$

because we are still observing the rigid motion of a plane. In other words, the optical flow is *form invariant*. Here, the old coordinates x,y and the new coordinates x',y' are related by

$$\begin{bmatrix} x' \\ y' \end{bmatrix}=\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}. \tag{D.2}$$

Since the velocity components are transformed as a vector, the old components u, v and the new components u', v' are also related by

$$\begin{bmatrix} u' \\ v' \end{bmatrix}=\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}. \tag{D.3}$$

If we substitute eqns (D.2) and (D.3) into eqns (D.1) and compare the result with eqns (2.1), we find that u_0, v_0 are transformed as a vector, A, B, C, D are transformed as a tensor, and E, F are transformed as a vector, namely,

$$\begin{bmatrix} u_0' \\ v_0' \end{bmatrix}=\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} u_0 \\ v_0 \end{bmatrix}, \tag{D.4}$$

$$\begin{bmatrix} A' & B' \\ C' & D' \end{bmatrix}=\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} A & B \\ C & D \end{bmatrix}\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{D.5}$$

$$\begin{bmatrix} E' \\ F' \end{bmatrix}=\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} E \\ F \end{bmatrix}. \tag{D.6}$$

Eqns (D.4), (D.5) and (D.6) are a linear mapping from u_0, v_0, A, B, C, D, E, F to u_0', v_0', A', B', C', D', E', F', and this mapping is a *representation*, i.e., a homomorphism, of the 2D rotation group. As is well known in group representation theory, any representation is reduced to one-dimensional *irreducible representations* due to *Schur's lemma*, since the 2D rotation group is compact and Abelian. In fact, if we define U_0, T, R and S as eqns (2.3), the above mapping is rewritten as

$$U_0'=e^{-i\theta}U_0, \quad T'=T, \quad R'=R,$$
$$S'=e^{-2i\theta}S, \quad K'=e^{-i\theta}K. \tag{D.7}$$

As Herman Weyl pointed out, irreducible representations describe physical quantities which are inherent to the phenomenon and independent of the choice of the coordinate system. Indeed, the above parameters describe geometrical characteristics of the flow itself familiar in fluid dynamics as is stated in the text. In particular, T, R and S are obtained by resolving the matrix composed of A, B, C and D into the scalar part, the deviator (or traceless symmetric) part and the antisymmetric (or skew) part. This is not a coincidence; according to the general theorem of Weyl, *all* irreducible representations of any *tensor representation* of $SO(n)$ are obtained by a combination of these decomposition processes.

GC970145-17

ROBUST ESTIMATION OF 3-D MOTION PARAMETERS FROM A SEQUENCE OF IMAGE FRAMES USING REGULARIZATION[1]

Gerard Medioni and Yoshio Yasumoto[2]

Intelligent Systems Group
Electrical Engineering
PHE 526 mc 0273
University of Southern California
Los Angeles, CA 90089-0273

ABSTRACT

In this study, we look at the issue of accurate estimation of the 3-D motion parameters of a rigid body from a sequence of synthetic images, and relate the effect of some parameters to the shape of an error function. We first consider the case where only a small set of corresponding points is identified and suggest that a technique called **regularization** improves the quality and stability of a solution. We then observe that, if more pairs of corresponding points are available, the error function becomes smooth and the solution stable. Finally, we try to improve the quality of estimation by considering more than 2 consecutive frames for a moving camera looking at a stationary scene, and summing the error functions obtained for any 2 consecutive frames. Surprisingly enough, this technique does not improve stability unless we use regularization again.

Keywords: Dynamic scene analysis, image sequence analysis, motion estimation.

1. INTRODUCTION

In recent years, many studies on the estimation of the 3-D motion of a rigid body have been performed. Some early works made simplifying assumptions, such as rotation around a fixed axis [22], [17], [3], orthographic projection [20], translational motion only [5, 10]. In most formalisms, the authors are led to solving a set of non linear equations, recently Tsai and Huang [19] and Longuet-Higgins [11] independently obtained closed form solutions and a set of linear equations.

The general paradigm for time-varying imagery analysis is as follows:

- feature extraction
- feature matching
- motion parameters estimation (and depth recovery)

In the formalism of optical flow, the first 2 steps are merged in the computation of the optical flow. If the image contains two or more objects moving independently, a segmentation procedure becomes necessary. In this study, we only look at the third step, the estimation of the 3-D motion parameters for a single rigid object, and our concern is the applicability to *real world* images, even though we only present results on synthetic imagery so far. Very few authors reported on such experiments: Dreschler and Nagel [6] only estimated image displacement, Roach and Aggarwal [15] used TV images but no results were given; finally Fang and Huang [7] used feature points with subpixel accuracy, but the results are not very accurate.

The ability to process real world images implies the ability to cope with some amount of noise in the input data, such as noise from the sensors, the digitization and feature extraction processes.

In that respect, the formulation developed by Tsai and Huang [19] is not appropriate, as they report 54% error in the estimation of the translation with 8 corresponding points for 1% perturbation of the input. Bruss and Horn [4] use the least square criterion to minimize the difference between measured and predicted displacement. This technique is harder to implement, but provides some tolerance to noise. Similarly, Adiv [1] uses a modified least square error function to obtain the value of the translation component, and then solves a set of linear equations to find the rotation vector. Here, we identify the problem of estimating 3-D motion as an *ill-posed inverse problem*. This suggests the applications of a general technique called regularization developed by russian mathematicians in the last twenty years [2, 18], and suggested by Poggio [14] as a natural mechanism in early vision.

[1]This research was supported, in part, by the Defense Advanced Research Projects Agency and was monitored by the Air Force Wright Aeronautical Laboratories under contract F33615-84-1404, Darpa order no. 3119.

[2]Yoshio Yasumoto is an Engineer for Matsushita Electric Industrial Co., Ltd. and was a visiting scholar at USC.

The next section introduces the ideas of regularization, section 3 gives the notations and conventions, section 4 shows the influence of the regularization factor when only 8 corresponding pairs are known, with and without quantization noise. Section 5 shows that if more pairs are known, then the error function is smooth and therefore regularization does not improve the result. Section 6 presents the application of the technique to 5 consecutive frames with constant motion parameters, and we express our conclusions in section 7

2. REGULARIZATION

The formulation of many practical problems leads to ill-posed problems By definition, a problem is *well-posed* if:

- There exists a solution
- the solution is unique
- the dependence of the solution on the input data is continuous:

From this definition, we can see that the problem of estimation of 3-D motion parameters is ill-posed since the solution is not robust in the presence of noise, and is not unique. furthermore. it is an inverse problem, and most inverse problems are ill-posed

Regularization proposes to "solve" these problems by restricting the space of acceptable solutions by imposing additional constraints

In general, for inverse problems, one of the formulation with regularization is as follows
Let our input data be y We are looking for a solution z such that $Az=y$ To do that, we choose a norm $\|\|$ and a *stabilizing function* Pz We then reformulate the problem as:
find a function z that minimizes $\|Az-y\|^2 + \lambda\|Pz\|^2$

The first term expresses the closeness of the solution to the input data, the second expresses the degree of regularization, or the additional constraints, and the factor λ controls the compromise between these two terms

In our problem, we wish to incorporate the fact that Ω and T should not change wildly when the input data are slightly perturbed. The simplest functional would therefore be of the form

$$\|Pz\|^2 = \lambda_1\Omega_x^2 + \lambda_2\Omega_y^2 + \lambda_3\Omega_z^2 + \mu_1T_x^2 + \mu_2T_y^2 + \mu_3T_z^2$$

The translation factor, however, is only defined up to a scale factor, making it difficult to include T in the regularization term. For simplicity, we also choose $\lambda_1=\lambda_2=\lambda_3$, therefore applying the regularization on the length of the vector Ω

Therefore $Pz = \sqrt{\Omega^2_x + \Omega^2_y + \Omega^2_z}$

Intuitively, the explanation is as follows: One of the assumptions is that the rotation angles are small enough to make the approximation $\sin\alpha=\alpha$, but the solving procedure does not check this condition. The new functional can be regarded as: find the solution minimizing the error, subject to the constraint that the vector Ω is small. The second term is then a Lagrange multiplier expressing the constraint.

3. NOTATIONS AND CONVENTIONS

In the object-coordinate frame, the following is the fundamental equation for a moving rigid object.

$$(x' y' z')^T = R\ (x y z)^T + T \qquad (1)$$

where $(x\ y\ z)$ denotes the object-space coordinates of a point P before motion and $(x'\ y'\ z')$ denotes the object-space coordinates of P after motion and R is a 3 X 3 orthonormal matrix of the first kind whose elements are expressed as a function of the rotation angle θ, and of the axis of rotation n1, n2, and n3, and $T=(T_x, T_y, T_z)^T$.

If we assume small rotation angles, matrix R is expressed by:

$$R = \begin{bmatrix} 1 & -\Omega_z & \Omega_y \\ \Omega_z & 1 & -\Omega_x \\ -\Omega_y & \Omega_x & 1 \end{bmatrix} \qquad (2)$$

where Ω_x, Ω_y, Ω_z denote the rotation angle around the x, y and z axis respectively as shown in Figure 1.

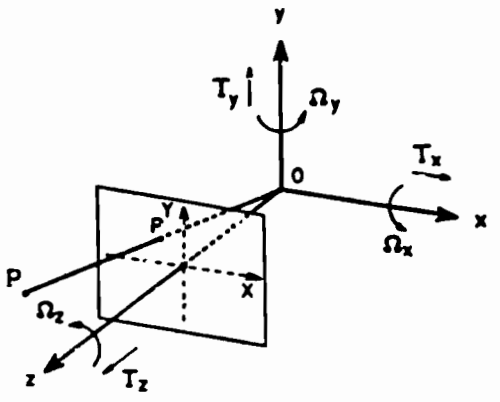


Figure 1: Coordinate System of Camera and Image Plane

The relations between object-coordinates and image-coordinates are as follows:

$$X = fx/z \qquad X' = fx'/z' \tag{3-1}$$

$$Y = fy/z \qquad Y' = fy'/z' \tag{3-2}$$

where f denotes the focal length and (X,Y) (X',Y') correspond to the projection of a point onto the image plane before and after motion.

Now, let (α, β) be the displacement vector (X'-X, Y'-Y). Using expressions (1) through (3), it is possible to derive an analytical expression of this vector at each point as a function of **R**, **T** and z

$$\alpha = \frac{-\Omega_x xy + \Omega_y(f^2+x^2) - \Omega_z fy + (T_x f - T_z x)f/z}{f - \Omega_x y - \Omega_y x + T_z f/z} \tag{4-1}$$

$$\beta = \frac{-\Omega_x(f^2+y^2) + \Omega_y xy + \Omega_z fx + (T_y f - T_z y)f/z}{f - \Omega_x y - \Omega_y x + T_z f/z} \tag{4-2}$$

By algebraic manipulation, and with no approximation, we get[3]

$$\alpha = -\Omega_x \frac{xy}{f} + \Omega_y\left(f + \frac{x^2}{f}\right) - \Omega_z \frac{y}{} + \frac{T_x f - T_z x}{z} \tag{5-1}$$

$$\beta = -\Omega_x\left(f + \frac{y^2}{f}\right) + \Omega_y \frac{xy}{f} + \Omega_z \frac{x}{} + \frac{T_y f - T_z y}{z} \tag{5-2}$$

By replacing as follows

$$\alpha_R = -\Omega_x \frac{xy}{f} + \Omega_y\left(f + \frac{x^2}{f}\right) - \Omega_z y \tag{6-1}$$

$$\alpha_T = \frac{T_x f - T_z x}{z} \tag{6-2}$$

$$\beta_R = -\Omega_x\left(f + \frac{y^2}{f}\right) + \Omega_y \frac{xy}{f} + \Omega_z x \tag{6-3}$$

$$\beta_T = \frac{T_y f - T_z y}{z} \tag{6-4}$$

We decompose the motion into independent rotation and translation terms.

$$\alpha = \alpha_R + \alpha_T \tag{7-1}$$

$$\beta = \beta_R + \beta_T \tag{7-2}$$

We measure the quantity (α, β) at each point and try to derive Ω_x, Ω_y, Ω_z, **T** and z.

As explained in the previous section, we choose the function to be minimized as a sum of 2 terms: the first one is the error measured in the least square sense, and the second term is the regularizing factor.

$$\sum_{i=1}^{n}[(\alpha_i - \alpha_{R_i} - \alpha_{T_i})^2 + (\beta_i - \beta_{R_i} - \beta_{T_i})^2] + \lambda(\Omega_x^2 + \Omega_y^2 + \Omega_z^2) \tag{8}$$

where α_i and β_i denote the displacement vector for the i^{th} point, which can be obtained from the consecutive images at each point, the first and second term are the square of the difference between the predicted and computed displacement, and the third term is the regularization function.

The problem is to obtain three rotation angles Ω_x Ω_y Ω_z, the translation vector, **T**, and the depth z at each point.

It is impossible to derive the absolute values of the translation vector **T** and of the depth z, but if the translation vector has length r not equal to 0, we can obtain the direction of **T**. Let us introduce a unit translation vector **U** and the relative depth d_i.

$$(U_x, U_y, U_z) = (T_x, T_y, T_z)/r \tag{9-1}$$

$$d_i = r/z_i, \qquad i = 1, \ldots, n \tag{9-2}$$

We can then rewrite equation (8) as follows:

$$\sum_{i=1}^{n}[(\alpha_i - \alpha_{R_i} - \alpha_{U_i}d_i)^2 + (\beta_i - \beta_{R_i} - \beta_{U_i}d_i)^2] + \lambda(\Omega_x^2 + \Omega_y^2 + \Omega_z \tag{10}$$

where

$$\alpha_{U_i} = U_x f - U_z X_i = \alpha_{T_i}/d_i \tag{11-1}$$

$$\beta_{U_i} = \text{''} \quad U_z Y_i = \beta_{T_i}/d_i \tag{11-2}$$

Thus, we try to compute the rotation angles, the unit translation vector U and the relative depth d_i at each point.

We can eliminate d_i by taking the first derivative of equation (10) with respect to d_i.

The derivative is

$$2[-(\alpha_i - \alpha_{R_i})\alpha_{U_i} - (\beta - \beta_{R_i})\beta_{U_i} + (\alpha_{U_i}^2 + \beta_{U_i}^2)d_i] \tag{12}$$

Setting this derivative to zero, we have:

$$d_i = \frac{(\alpha_i - \alpha_{R_i})\alpha_{U_i} + (\beta_i - \beta_{R_i})\beta_{U_i}}{\alpha_{U_i}^2 + \beta_{U_i}^2} \tag{13}$$

unless $\alpha_{U_i}^2 + \beta_{U_i}^2 = 0$

We have to take into account the constraint that all d_i should be positive, since all objects are on one side of the camera. If even one point is negative, it may be appropriate to assume that d_i should be zero [1]. In the searching procedure mentioned later, we eliminate values of the parameters leading to points with negative d_i by checking this condition of each point for i

[3] This equation is different from [1], in which $T_z << z$ and Ω_x, Ω_y small was required.

We therefore evaluate expression δ_i which is the numerator of equation (13).

$$\delta_i=(\alpha_i-\alpha_{Ri})\alpha_{Ui}+(\beta_i-\beta_{Ri})\beta_{Ui} \qquad (14)$$

Now we can write the error function E_i for each corresponding pair by replacing d_i in (10) by its expression (13), and after manipulation, we have:

$$E_i=\frac{[(\alpha_i-\alpha_{Ri})\beta_{Ui}-(\beta_i-\beta_{Ri})\alpha_{Ui}]^2}{\alpha_{Ui}^2+\beta_{Ui}^2} \qquad \text{if } \delta_i>0 \quad (15)$$

The function to be minimized $E(U,\Omega)$ as a function of only the motion parameters is:

$$E(U,\Omega)=\sum_{i=1}^{n}E_i+\lambda(\Omega_x^2+\Omega_y^2+\Omega_z^2) \qquad (16)$$

This function is defined only when all δ_i's are positive. We now have to search the space to find the minimum of this error function. As we choose a unit translation vector, we can simply search the surface of the sphere with unit radius. Furthermore, equation (16) returns the same value for symmetric points on the sphere, so we only have to search the surface of one hemisphere and check condition (14) for both points.

Given a point on the hemisphere, three linear equations as a function of Ω_x, Ω_y, Ω_z are obtained by taking the partial derivative with respect to them.

$$\begin{aligned}
\frac{\partial E(U,\Omega)}{\partial \Omega_x} &= \sum_{i=1}^{n}\frac{\partial E_i}{\partial \Omega_x}+2\lambda\Omega_x = 0\\
\frac{\partial E(U,\Omega)}{\partial \Omega_y} &= \sum_{i=1}^{n}\frac{\partial E_i}{\partial \Omega_y}+2\lambda\Omega_y = 0 \qquad (17)\\
\frac{\partial E(U,\Omega)}{\partial \Omega_z} &= \sum_{i=1}^{n}\frac{\partial E_i}{\partial \Omega_z}+2\lambda\Omega_z = 0
\end{aligned}$$

From these linear equations, the three rotation angles are obtained. Using these three rotation angles, the value of the error function (16) is obtained at each point on the hemisphere.

Since we are searching on a half hemisphere representing the possible values of the (normalized) translation vector, we can parameterize this space with 2 bounded variables, θ and ϕ and express any unit vector $U = (U_x, U_y, U_z)^T$ as:

$$\begin{aligned}
U_x&=\sin\phi\cos\theta\\
U_y&=\sin\phi\sin\theta \qquad (18)\\
U_z&=\cos\phi
\end{aligned}$$

So we use a cartesian system in which the "x" axis represents θ and the "y" axis represents ϕ. The two level search procedure is as follows:

1) Coarse search: We first quantize the θ, ϕ cartesian into a 36 by 36 array corresponding to 10° steps along θ and 2.5° steps along ϕ. For each cell, we compute the function $E(U,\Omega)$ and check that all δ_i have the same sign.

2) Fine search: In the cell(s) in which the minimum occured, we perform a step by step search in 1° increments for both axes.

Figure 2 shows the shape of the error function obtained by the first stage, which is described in the next section, and in black below it the regions where a solution is admissible, that is where condition (14) is met. These drawings correspond to the first data set of table 1.

4. 3-D MOTION PARAMETERS WHEN 8 POINTS ARE MATCHED

In this section, we generate synthetic data by choosing a set of 3-D points, inputting the translation vector, the rotation vector and the focal length of the camera and generating a set of 2-D corresponding points in the image plane. The coordinates of these points are real numbers, but we can simulate digitization noise by using the truncated corresponding integer values, or by restricting the number of significant digits.

DATA 1				DATA 2			
X	Y	X'	Y'	X	Y	X'	Y'
166.83163	134.62000	166.99232	134.81388	282.56746	182.86714	254.67588	166.33000
43.885714	31.885714	47.811562	29.734764	147.12644	147.12644	139.37811	131.87700
64.000000	32.000000	67.300829	31.188298	213.33333	166.66666	197.86466	142.72794
88.711111	83.811111	88.883292	84.272936	178.66667	128.00000	169.86754	116.82963
72.891366	30.986173	75.889466	28.167529	287.75194	183.46837	254.82132	166.83266
142.66662	128.66667	144.44829	123.28666	176.39436	158.66670	166.86087	143.31827
116.66666	120.66646	116.63441	121.12139	166.67442	146.66647	160.66662	131.74316
252.1212°	61.676767	267.21667	58.266201	138.63636	127.81818	133.24666	121.66286
Focal Length = 128				Focal Length = 128			
θ_x=1.4° θ_y=1.5° θ_z=1.8°				θ_x=1.8° θ_y=0° θ_z=1.8°			
U_x=U_y=U_z				U_x=U_y=$U_z/2$			

Table 1: Point-Point Correspondences in 2 Synthetic Images

Table 1 shows 2 sets of 8 corresponding points which we use in this experiment. The rotation and translation parameters are given below the corresponding pairs. In order to interpret the results of the experiment, we need to display our error function in a meaningful and efficient manner which was explained in the previous section. For reasons of clarity, we do not display the error function E, but -E, so that we have to look at the maximum of -E (corresponding to the minimum of E). For all figures, a white arrow indicates the expected position of the extremum, and a dark arrow shows the computed position of the extremum.

For the first data set, the correct answer should be θ=225°, ϕ=55°. Figure 3 shows the shape of the error function for 3 different values of λ(0, 0.1, 0.5). We can see the smoothing effect of the regularization term on the error function. In all cases, the solution found is very close to the desired one, which is expected since our data is not noisy.

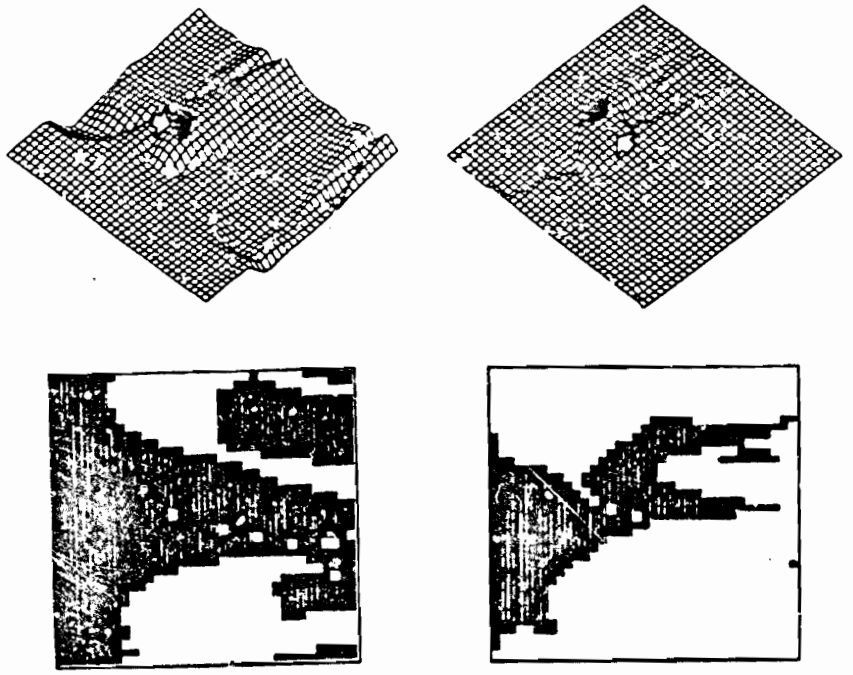


Figure 2: Error Function and The Space of Acceptable Solutions (DATA 1)

The black area shows the admissible space
(for which all points have positive z)

Table 2 shows the motion parameters obtained with the different values of λ on this first data set. For completeness, we also include the results obtained by the linear method of Tsai and Huang [19]. For that method, U_y is always 1.0, so that -U instead of U may be found. For real input, all methods give small error ranging from: 0 % to 5 % for the rotation parameters and from 0 % to 20 % for the translation parameters.

In order to understand the effects of noise, we convert the real numbers to the closest integer. Given the range of our data, the error introduced could be as large as 1.5%, with an average of less than 0.5%. This would be equivalent to choosing features in an image with pixel accuracy, which is common in "real world" images. Even with such a small amount of noise, the error function exhibits some peaks leading to erroneous motion parameters. The effect of the regularization factor on the shape of the error function is demonstrated in figure 3 for 3 values of the parameters λ(0.0, 5.0, 10.0). For λ=0.0 the searching procedure produces a spurious maximum far from the expected value. As we increase the value of λ the maximum gets closer to the desired value. The results are summarized in table 2. In this table, for integer input, we observe that the error decreases as λ increases. The rotation parameters are very well estimated by the linear method, but not the translation parameters.

We repeat the procedure on a different data set shown in table 1 in which the Ω vector is not symmetric. For this example, the correct answer should be θ=45°, ϕ=35°, Ω_x=1.0°, Ω_y=0°, Ω_z=-1.0°. Using all significant digits of the input data and no regularization, we obtain: θ=45°, ϕ=35° (U_x=0.5, U_y=0.5, U_z=1.0) and the associated rotation vector Ω_x=0.99°, Ω_y=-0.01°, Ω_z=-1.01°.

If we now use integer values for input, we find the totally incorrect parameters θ=155°, ϕ=37°. These results are to be compared with the parameters obtained by searching the space for an error function including a regularization factor with λ=5.0. The obtained results including linear method [19] are summarized in table 3. In this example, both the linear and no-regularization methods give quite incorrect parameters for translation.

Real Input

Method	linear	Error	λ=0.0	Error	λ=0.1	Error	λ=0.5	Error
Ω_x(°)	1.37	-0.03	1.40	0.00	1.41	0.01	1.45	0.05
Ω_y(°)	1.53	0.03	1.50	0.00	1.48	-0.02	1.43	-0.07
Ω_z(°)	1.58	0.02	1.60	0.00	1.60	0.00	1.59	-0.01
U_x	-1.00	0.00	1.01	0.01	0.84	-0.08	0.83	-0.17
U_y	-1.00	0.00	1.01	0.01	0.84	-0.08	0.80	-0.20
U_z	1.00	-	-1.00	-	-1.00	-	-1.00	-
θ(°)	-	-	225	-	225	-	224	-
ϕ(°)	-	-	55	-	53	-	48	-

Integer Input

Method	linear	Error	λ=0.0	Error	λ=5.0	Error	λ=10.0	Error
Ω_x(°)	1.24	-0.16	4.06	2.66	3.21	1.81	2.39	0.99
Ω_y(°)	1.82	0.32	0.75	-0.75	0.95	-0.55	1.30	-0.50
Ω_z(°)	1.61	0.01	0.06	-1.54	0.97	-1.03	1.13	-0.47
U_x	0.45	-1.45	0.11	-0.89	0.13	-0.87	0.28	-0.74
U_y	0.39	-1.39	2.04	1.04	1.88	0.88	1.21	0.21
U_z	1.00	-	-1.00	-	-1.00	-	-1.00	-
θ(°)	-	-	247	-	268	-	256	-
ϕ(°)	-	-	64	-	[illegible]	-	51	-

Input Data: θ=225° (ϕ=55° (U_x=U_y=-U_z)), Ω_x=1.4°, Ω_y=1.5°, Ω_z=1.6°

Table 2: The Motion Parameters (DATA 1)

and rotation. With regularization, the value is not absolutely correct, but stays close to the expected value, even though the input is not very accurate. The estimated amounts of noise in the input is between 0.35 and 0.9 %. The corresponding error functions are displayed on figure 4.

DATA 2 is a typical example demonstrating the improvement due to regularization. The non-regularization and the linear method [19] perform very poorly here. We generated many examples with many combinations of small rotation angles and any translation parameters, and where 10 to 30 pixels displacements were observed on the image plane. This assumption is reasonable for camera looking at an object off center. We found that, with no exceptions, regularization vastly improves the results.

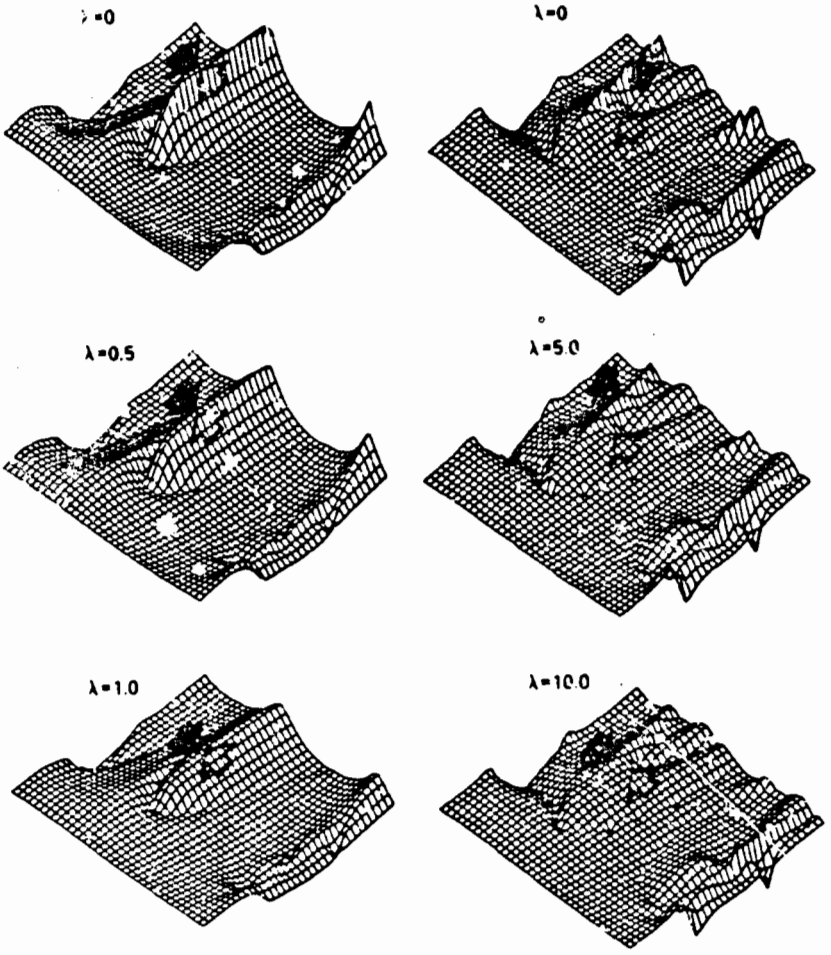


Figure 3: Error function of DATA 1

Input Data: $\theta=225°$ $\phi=55°$ ($U_x=U_y=-U_z$), $\Omega_x=1.4°$ $\Omega_y=1.5°$ $\Omega_z=1.6°$

Real Input

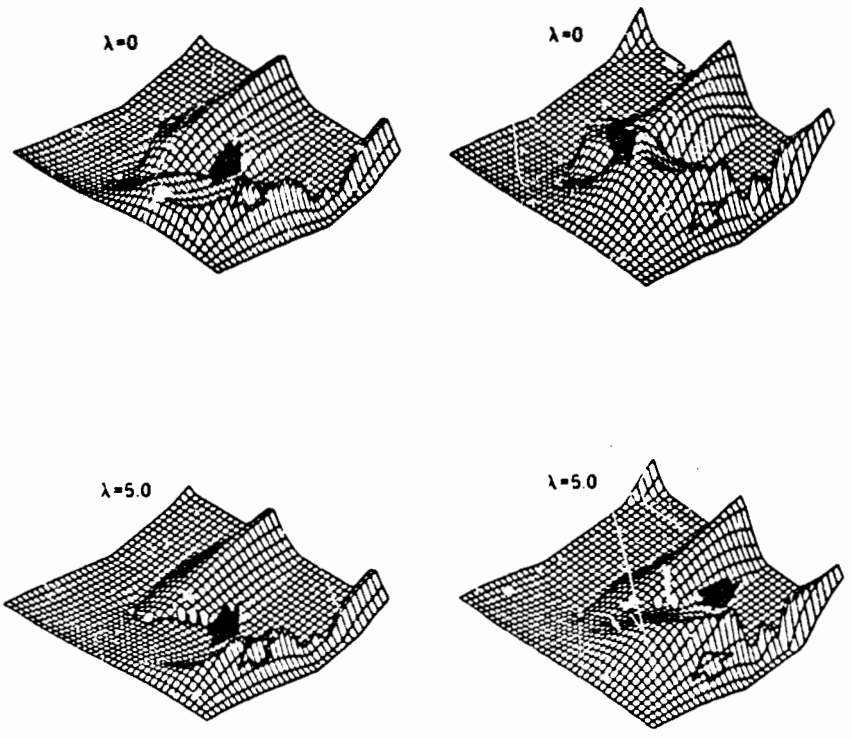
	Answer	Error	λ=5.0	Error	λ=5.0	Error
Ω_x(°)	1.00	0.00	0.99	−0.01	0.87	−0.13
Ω_y(°)	−0.00	0.00	−0.01	−0.01	0.20	0.20
Ω_z(°)	−1.00	0.00	−1.01	−0.01	−1.00	0.00
U_x	0.50	0.00	0.5	0.00	0.43	−0.07
U_y	0.50	0.00	0.5	0.00	0.45	−0.05
U_z	1.0	-	1.0	-	1.0	
θ(°)	-	-	45	-	45	-
φ(°)	-	-	35	-	32	-

Integer Input

	Answer	Error	λ=0.0	Error	λ=5.0	Error
Ω_x(°)	−3.00	−4.00	−8.36	−7.36	1.19	0.19
Ω_y(°)	−1.30	−1.30	20.7	20.7	−0.87	−0.87
Ω_z(°)	−25.2	−24.2	−13.3	−12.3	−1.83	−0.83
U_x	1.19	0.00	−0.00	−1.19	0.00	0.19
U_y	−1.49	−1.30	0.32	−0.16	0.00	0.30
U_z	1.0	-	1.0	-	1.0	
θ(°)	-	-	195	-	33	-
φ(°)	-	-	37	-	46	-

Input Data: θ=45° φ=35° (U_x=U_y=U_z/2), Ω_x=1.0°, Ω_y=0°, Ω_z=−1.0°

Table 3: The Motion Parameters (DATA 2)



Real Input **Integer Input**

Figure 4: Error function for DATA 2

Input Data: θ=45° φ=35° (U_x=U_y=U_z/2), Ω_x=1.0°, Ω_y=0°, Ω_z=−1.0°

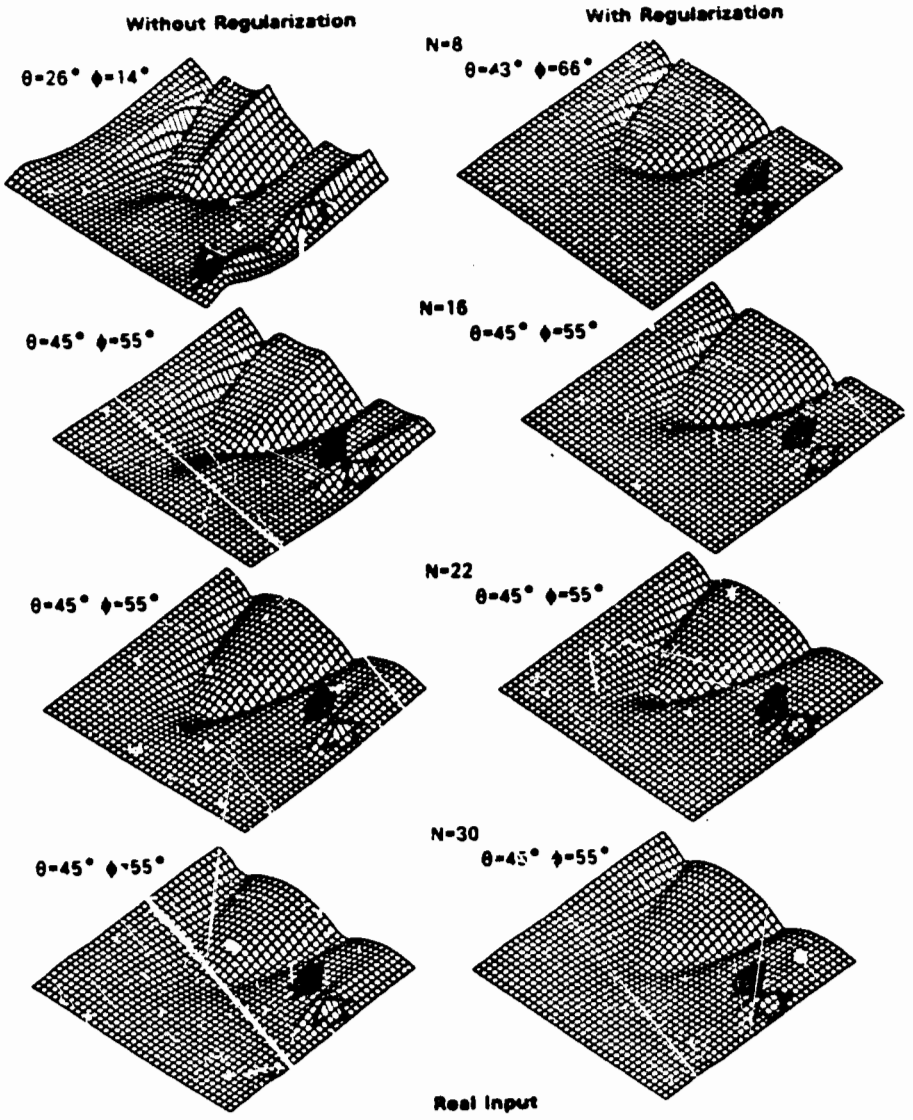


Figure 5: Error function as the number of points increases

Input Data. $\theta=45°$ $\phi=55°$ ($U_x=U_y=U_z=1$), $\Omega_x=12°$ $\Omega_y=13°$ $\Omega_z=14°$

5. MORE THAN 8 CORRESPONDING POINTS

In this section, we take a look at the way the error function varies as a function of the number of corresponding points. To obtain the corresponding points, we proceed as previously by taking a random collection of 3-D points, choosing the motion parameters and computing the resulting associated pairs. The error function is displayed in Figure 5, for the values N=8, 16, 22 and 30 of the number of corresponding pairs, with and without the regularization term.

As we can clearly see, the more points we take into account, the smoother the error function becomes, and if we use a regularizing factor, the error function is already smooth for N=8, and in this case more accurate value can be obtained than error function without regularization.

This indicates that the effects of the regularization are especially useful when only a few pairs are known.

6. MORE THAN 2 CONSECUTIVE FRAMES

Many researchers have proposed using three (or more) consecutive frames for motion analysis [20, 21, 9, 13]. Human perception of motion seems to be very noise sensitive when only 2 frames of moving dot patterns are presented [8].

Here, we create 5 consecutive images in which the motion parameters stay constant, and generate 4 sets of corresponding pairs. In this example, we assume that the camera is moving with constant parameters and that the scene stands still. This assumption is reasonable if images are obtained from a camera attached to a robot manipulator moving with constant rotation and translation parameters. With these assumptions we can write for any point $M^{(i)}$ in frame i,

$$OM^{(i+1)} = R\, OM^{(i)} + T \qquad (19)$$

We compute the error function for the 4 consecutive frame pairs. All data are truncated to the closest integer, in order to simulate noise, and the amount depends on the value of coordinates on the image plane. In this example, from frame 1-2 through 4-5, the amount of noise is 0.2 - 0.4 %, 0.3 - 0.6 %, 0.3 - 0.8%, and 0.3 - 1.2 %, respectively. Although we use the same motion parameters, the computed values vary between frames, as shown on figure 6, where the right column represents the error function with regularization (λ=5.0) and the left one the error function without regularizing factor (λ=0.0). The last row shows the error function obtained by averaging the 4 previous error functions. In table 4, we show the values obtained from these image sequences, including the results of the linear method [19].

The computed extrema of the error function are very similar for the 5 inter-frames when we use regularization, which is not true when λ=0.0. We observe that computed errors without regularization increase according to the errors included in images. As shown on the last row of the figure, searching the sum of the 4 error functions leads to a nearly correct extremum **only with the regularization term**.

This serves to demonstrate that smoothing the error function by simply averaging it over a time sequence is not enough to overcome the fact that the problem is ill-posed.

	Linear	Error	Frames No. 1 - 2 λ=0.0	Error	λ=5.0	Error
θ_x(°)	0.00	0.00	11.7	10.7	0.26	-0.72
θ_y(°)	1.15	1.15	-1.00	-1.00	0.40	0.40
θ_z(°)	4.72	-2.72	-13.4	-11.4	-1.00	0.40
U_x	1.34	0.34	-0.00	-1.00	0.00	-0.12
U_y	1.12	1.62	2.00	3.3	-0.55	-1.05
U_z	1.6	0.00	-1.0	-2.06	1.0	0.00
θ(°)	-	-	282	-	328	-
ϕ(°)	-	-	71	-	48	-
	Linear	Error	Frames No. 2 - 3 λ=0.0	Error	λ=5.0	Error
θ_x(°)	5.32	4.32	-25.5	- 30.5	-1.00	-2.00
θ_y(°)	0.16	0.16	2.30	2.30	0.00	0.00
θ_z(°)	-4.00	-2.00	27.5	29.5	1.00	3.00
U_x	1.15	0.15	0.05	-0.05	0.00	0.01
U_y	0.01	1.31	-1.72	-1.22	-0.00	-0.40
U_z	1.0	0.00	1.0	0.00	1.0	0.00
θ(°)	-	-	269	-	316	-
ϕ(°)	-	-	63	-	64	-
	Linear	Error	Frames No. 3 - 4 λ=0.0	Error	λ=5.0	Error
θ_x(°)	5.53	4.53	21.5	30.5	0.33	-0.67
θ_y(°)	0.40	0.40	0.67	1.07	0.12	0.12
θ_z(°)	-3.47	-1.47	-26.3	-24.3	-0.02	1.30
U_x	1.13	0.13	-1.0	-2.0	1.0	0.00
U_y	0.50	1.00	0.1	0.5	-0.05	-0.15
U_z	1.0	0.00	-1.0	0.00	1.0	0.00
θ(°)	-	-	286	-	328	-
ϕ(°)	-	-	81	-	51	-
	Linear	Error	Frames No. 4 - 5 λ=0.0	Error	λ=5.0	Error
θ_x(°)	4.11	3.11	2.70	1.70	-0.03	-1.03
θ_y(°)	-0.05	-0.05	-0.35	-0.35	-0.42	-0.42
θ_z(°)	-3.52	-1.52	-4.22	-2.22	-0.97	1.03
U_x	1.05	0.05	1.10	0.10	1.10	0.10
U_y	2.40	0.05	0.10	0.00	-1.00	-0.90
U_z	1.0	0.00	1.0	0.00	1.0	0.00
θ(°)	-	-	16	-	310	-
ϕ(°)	-	-	46	-	67	-
	Linear	Error	Approximation λ=0.0	Error	λ=5.0	Error
θ_x(°)	5.47	4.47	10.46	9.46	-1.41	2.41
θ_y(°)	0.44	0.44	1.04	1.04	0.10	0.10
θ_z(°)	4.00	4.00	-16.42	-14.42	-2.11	-0.11
U_x	1.32	0.32	1.07	0.07	1.00	0.00
U_y	0.75	0.25	-1.09	-1.34	-0.84	-0.34
U_z	1.00	0.00	1.00	0.00	1.00	0.00
θ(°)	-	-	300	-	326	-
ϕ(°)	-	-	66	-	53	-

Integer Input

Input Data: θ=33.3° ϕ=46° $(U_x=-2U_y=U_z)$ θ_x=1°, θ_y=0°, θ_z=2°

Table 4: The Motion Parameters (Consecutive Images)

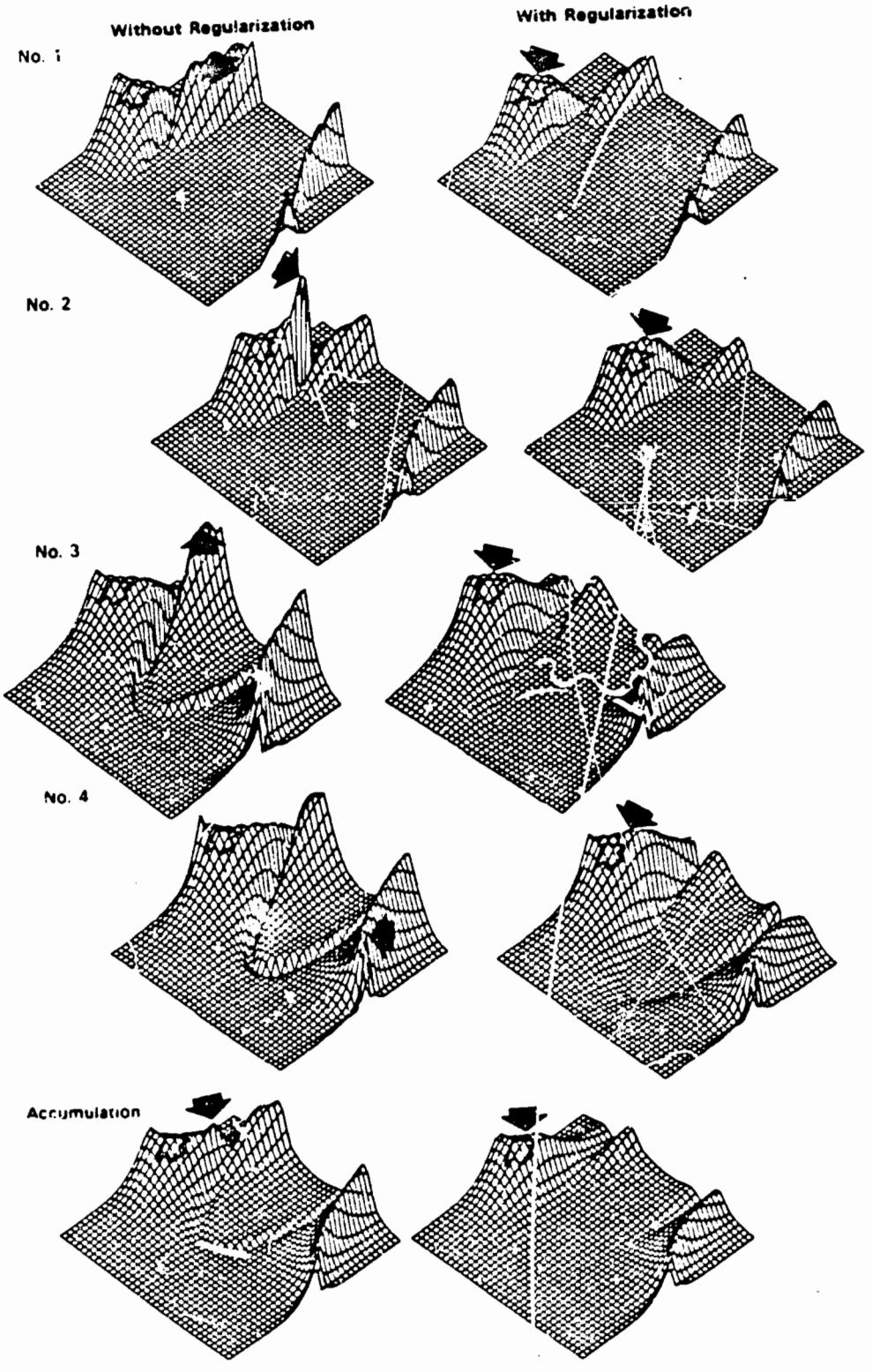


Figure 6: Error function for five consecutive images

Input Data: θ=333° ϕ=48° (U_x=-2U_y=U_z), Ω_x=10°, Ω_y=0°, Ω_z=-20°

7. CONCLUSION AND FUTURE RESEARCH

In this experimental study, we have made clear the usefulness of a regularizing term in at least 2 instances.

- In the presence of noise

- When only a few points are known

We found that such a technique stabilizes the value of the desired parameters in a "noisy" image, and may be applicable in pixel-based image analysis and real time motion analysis.

One problem that has not been examined here is the optimal choice of the parameter λ. It obviously depends on the amount of noise in the image and could be automatically set if a valid model of both image and noise were possible to derive, as in a previous work [12].

Our final remarks relate to the shape of the error function, which is nearly flat on large portions of the search space, as noted by [1], making it very difficult to select a extremum. This seems to suggest that the formulations of the problem used so far do not capture enough information, or do not put enough restrictions on the solution. A different approach is needed, where the solution would appear as a **sharp** peak. One possibility being currently investigated is to apply image plane *acceleration* to motion analysis [16].

ACKNOWLEDGMENTS

We would like to thank Michel Medioni for deriving equation (5), and Hormoz Shariat for fruitful discussions.

References

[1] Adiv, G.
Determining 3-D Motion and Structure from Optical Flow Generated by Several Objects.
In *Proceedings of Image Understanding Workshop October 1984*, pages 113-129. DARPA, Science Application International Corporation, 1984.

[2] Arsenin, V. Ya.
Regularization Method.
USSR Comp. Math. 8, 1968.

[3] Bobick, A.
A Hybrid Approach to Structure-from-Motion.
In *Proc. ACM Interdisc. Workshop on Motion, Toronto*, pages 91-109. 1983.

[4] Bruss, A.R. and Horn, B.K.P.
Passive Navigation
Massachusetts Institute Technology A.I. Memo 662 (662), 1981

[5] Clocksin, W.F.
Perception of surface slant and edge labels from optical flow: A Computational approach.
Perception, 1980.

[6] Dreschler, L. and Nagel, H.-H.
Volumetric Model and 3-D trajectory of a Moving Car Derived from Monocular TV-frame Sequence of a Street Scene.
In *Proceedings of 7th IJCAI, Vancouver, Canada*. August, 1981

[7] Fang, J.-Q. and Huang, T.S.
Some Experiments on Estimating the 3-D Motion Parameters of a Rigid Body from Two Consecutive Image Frames
IEEE Trans. on Pattern Analysis and Machine Intelligence 6(5) 545-554, 1984.

[8] Lappin, J.S., Doner, J.F., and Kottas, B.L.
Minimal Conditions for the Visual Detection of Structure and Motion in Three Dimensions.
Science 209 717-719, 1980

[9] Lawton, D.T.
Constraint-Based Inference from Image Motion
In *Proceedings of the First Annual National Conference on Artificial Intelligence*. August, 1980

[10] Lee, D.N.
The Optical Flow Field: The foundation of vision
Phil. Trans. Royal Soc., London, 1980

[11] Longuet-Higgins, H.C.
A Computer Algorithm for Reconstructing a Scene from Two Projections.
Nature, 1981

[12] Medioni, G. and Moriamez, P.
Tomographie. Reconstruction d'Images par Deconvolution Numerique et Optique dans le Plan de Fourier.
Tech. Report ENST-H-77002, Ecole Nationale Superieure Des Telecommunications, June, 1977.

[13] Meiri, A.Z.
On Monocular Perception of 3-D Moving Objects.
IEEE Trans. on Pattern Analysis and Machine Intelligence 2:582-583, 1980.

[14] Poggio, T. and Torre, V.
Ill-posed Problems and Regularization Analysis in Early Vision.
In *Proceedings of Image Understanding Workshop October 1984* DARPA, Science Application International Corporation, 1984.

[15] Roach, J.W., and Aggarwal J.K.
Determining the Movement of Objects from a Sequence.
IEEE Trans. on Pattern Analysis and Machine Intelligence 2:554-552, 1980

[16] Shariat, H.
The Motion Problem: A Decomposition-Based Solution.
In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition at San Francisco*, pages 181-183, 1985

[17] Sugie, N. & inagaki, H.
A Computational Aspect of Kinetic Depth Effect.
Biol. Cybern. 50:431-436, 1984.

[18] Tikhonov, A.N.
The Regularization of ill-Posed Problems.
Dokl. Akad. Nau. SSR 153(1):49-52, 1963.

[19] Tsai, R.Y. and Huang, T.S.
Uniquenes: and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces.
IEEE Trans on Pattern Analysis and Machine Intelligence 6(1):13-26, January, 1984.

[20] Ullman, S.
The Interpretation of Visual Motion.
Massachusetts Institute Technology Press, 1979.

[21] Ullman, S.
Maximizing Rigidity: The Incremental Recovery of 3D Structure from Rigid and Non Rigid Motion.
Perception 13:255-274, 1984.

[22] Webb, J.A. and Aggarwal, J.K.
Structure from Motion of Rigid and Jointed Bodies.
In *Proceedings of 7th IJCAI, Vancouver, Canada*, 1981.

GC970145-18

Contour, Orientation and Motion

John Aloimonos, Anup Basu and Christopher M. Brown

Computer Science Department
University of Rochester
Rochester, New York 14627

Abstract

Intrinsic image calculation exploits constraints arising from physical and imaging processes to derive physical scene parameters from input images. After a brief review of a paradigmatic intrinsic image calculation, we turn to a new result that derives shape and motion from a sequence of patterned inputs. Experimental results are demonstrated for synthetic images.

1. Shape, Orientation, and Paraperspective

One of the first and best-known examples of intrinsic image calculation [Barrow & Tenenbaum 1978] is the recovery of shape from intensity [e.g. Horn 1978, Ikeuchi & Horn 1981]. Shape of a smooth surface is defined to be local surface orientation or viewer-centered relative depth, which are derivable from each other. Orientation is parameterized by the unit normal vector of the surface $\vec{n}(x,y)$, or by the projection of such vectors with $z>0$ from the origin onto the plane $z = 1$. This projection yields the popular (p,q) or gradient space representation commonly used in reasoning about line drawings. A polar version of (p,q) space is (slant, tilt) space. For example, the orientation of a plane that is tilted in an upward direction lies somewhere on the positive p axis, the farther away from the origin the greater the slant of the plane. At infinite slant the plane is edge-on to the viewer, with its normal perpendicular to the line of sight.

Ohta's [1981] perspective approximation (called paraperspective here) is useful. Let a coordinate system OXYZ be fixed with respect to the camera, with the -Z axis pointing along the optical axis, and O the nodal point of the eye (center of the lens). The image plane is assumed to be perpendicular to the Z axis at the point (0,0,-1), (i.e. focal length = 1). Ohta approximates perspective projection by a two-stage affine transformation. A texel is taken to lie on the tangent plane Q of the surface at the texel's centroid. Q has orientation (p,q). A plane P is erected through the texel centroid parallel to the image plane, and the texel's shape is projected, parallel to the line joining the viewpoint and the texel centroid, onto P. This first stage is a skew transformation. The second stage is a true point projection from plane P to the image plane through the viewpoint. Since P and the image plane are parallel, this projection amounts to a pure scaling by some constant factor, say $1/\beta$. Paraperspective approximates location- and depth-dependent perspective foreshortening and size distortion, and the approximations are quite good for small shapes.

To represent the original pattern of the surface texel, we use an (a,b,c) coordinate system, with its origin at the mass center of the texel and the (a,b) plane identical to the plane Q. To represent the pattern of the image texel, we use an (a',b',c') coordinate system, with its origin the point (A,B,-1), where (A,B) is the mass center of the image texel, and the axes a',b',c' are parallel to the axes X,Y,Z respectively. Then the transformation from (a,b) to (a',b') with the two step projection process of previous section is given by an affine transformation of equation (*). In our work we choose one P plane for the entire image, which implies that depth variation must be small relative to depth.

$$\begin{bmatrix} a' & b' \end{bmatrix} = \begin{bmatrix} a & b \end{bmatrix} \frac{1}{\beta} \begin{vmatrix} \frac{-1+pA}{\sqrt{1+p^2}} & \frac{pB}{\sqrt{(1+p^2)}} \\ \frac{q(p+A)}{\sqrt{(1+p^2)(1+p^2+q^2)}} & \frac{qB-p^2-1}{\sqrt{(1+p^2)(1-p^2+q^2)}} \end{vmatrix} \qquad (*)$$

This transformation expresses the relation between two 2-D patterns, one in the 3-D space and the other its image on the image plane. We now use it to develop a basic constraint.

2. The Area Ratio Constraint

The determinant of the matrix of an affine transformation is equal to the ratio of the areas of the two patterns before and after the transformation. Specifically, if S_W is the area of a world texel that lies on a plane with gradient (p,q) and S_I is the area of its image that has mass center (A,B), then we have:

$$\frac{S_I}{S_W} = \frac{1}{\beta^2} \det \begin{vmatrix} \frac{-1+pA}{\sqrt{1+p^2}} & \frac{pB}{\sqrt{(1+} } \\ \frac{q(p+A)}{\sqrt{(1+p^2)(1+p^2+q^2)}} & \frac{q(\cdot}{\sqrt{(1+p^2)(1+p^2+q^2)}} \quad \frac{1}{} \end{vmatrix}$$

or $$\frac{S_I}{S_W} = \frac{1}{\beta^2} \cdot \frac{1-Ap-Bq}{\sqrt{1+p^2+q^2}}$$

or $$S_I = \frac{S_W}{\beta^2} \cdot \frac{1-Ap-Bq}{\sqrt{1+p^2+q^2}} \qquad (**)$$

Equation (**) relates the area of a world texel S_W, its gradient (p,q), the area S_I of its image and its mass center (A,B). If we call the quantity S_I "textural intensity," and

the quantity S_W/β^2 "textural albedo," then equation (**) is very similar to the image irradiance equation for Lambertian surfaces:

$$I = \lambda \frac{1 + Ap + Bq}{\sqrt{1+p^2+q^2}}$$

where (p,q) is the gradient of the surface point whose image has intensity I, λ is the albedo at that point and (A,B,1) the direction of the light source (Horn, 1977; Ikeuchi, 1981). Thus equation (**) can be used to recover surface orientation. This shape from texture work is reported in [Aloimonos & Chou, 1985; Aloimonos & Swain, 1985].

3. Orientation from Contour Without Point Correspondence

Planar surface orientation may be recovered from contour information. In fact, the change of the perceived area of a planar contour from different cameras in a known configuration is enough to recover the 3-D structure of the contour, without the knowledge of point to point correspondence between the different images.

The recovery of three-dimensional shape and surface orientation from a two-dimentional contour is a fundamental process in any visual system. Recently, a number of methods have been proposed for computing this shape from contour. For the most part, previous techniques have concentrated on trying to identify a few simple, general constraints and assumptions that are consistent with the nature of all possible objects and imaging geometries in order to recover a single "best" interpretation, from among the many possible for a given image. For example, Kanade [1981] defines shape constraints in terms of image space regularities such as parallel lines and skew symmetries under orthographic projection. Witkin [1981] looks for the most uniform distribution of tangents to a contour over a set of possible inverse projections in object space under orthography. Similarly, Brady and Yuille [1984] search for the most compact shape (using the measure of area over perimeter squared) in the object space of inverse projected planar contours.

Rather than attempting to maximize some general shape-based evaluation function over the space of possible inverse projective transforms of a given image contour, we propose to find a unique solution by using more than one camera, since it can be easily proved that only one image (under orthography or perspective) of a planar contour admits infinite interpretations of the structure of the world plane on which the contour lies. Finally, the need for a unique solution, which is guaranteed in our approach, comes also from the fact that there exist many real world counterexamples to the evaluation functions that have been developed to date. For example, Kanade's and Witkin's measures incorrectly estimate surface orientation for regular shapes such as ellipses (which are often interpreted as slanted circles). Brady's compactness measure does not correctly interpret non-compact figures such as rectangles since he will compute it to be a rotated square (e.g. if we view a rectangular table top, we do not see it as a rotated square surface, but as a rotated rectangle.)

In the sequel, we present two methods for the unique recovery of shape from contour, one based on three views and the other based on two views, without having to solve the point to point correspondence between the different images of the contour. We proceed with the following proposition (Fig. 1).

Proposition: Let a coordinate system 0,X,Y,Z be fixed, with the -Z axis pointing along the optical axis. We consider that the image plane Im_1 is perpendicular to the Z axis at the point (0,0,-1). Let a plane Π with equation $-Z = pX + qY + c$ in the world, where (p,q) is the gradient of the plane that contains a contour C. Furthermore, we consider two more cameras with image planes Im_2 and Im_3, whose coordinate systems (nodal points) are such that any world point has the same depth with respect to any of the cameras. Then, assuming parperspective projection of the contour C onto any of the image planes, the images C_1, C_2, and C_3 of the contour on the three cameras are enough to determine uniquely the orientation of the plane Π, without having to solve the point to point correspondence between C_1, C_2, C_3.

Proof: Let S_1, S_2, and S_3 be the areas of the contours C_1, C_2 and C_3 respectively. Let also the depth of the center of gravity of the contour C be β. If S_W is the area of the contour C on the plane Π, and (A_1,B_1), (A_2,B_2) and (A_3,B_3) the centers of gravity of the image contours C_1, C_2 and C_3 respectively, then the area ratio constraint (**), is that:

$$\frac{S_1}{S_W} = \frac{1}{\beta^2} \frac{1 - A_1p - B_1q}{\sqrt{1+p^2+q^2}} \quad (1)$$

$$\frac{S_2}{S_W} = \frac{1}{\beta^2} \frac{1 - A_2p - B_2q}{\sqrt{1+p^2+q^2}} \quad (2)$$

$$\frac{S_3}{S_W} = \frac{1}{\beta^2} \frac{1 - A_3p - B_3q}{\sqrt{1+p^2+q^2}} \quad (3)$$

Dividing the above equations appropriately, we derive.

$$\frac{S_1}{S_2} = \frac{1 - A_1p - B_1q}{1 - A_2p - B_2q} \quad (4)$$

$$\frac{S_2}{S_3} = \frac{1 - A_2p - B_2q}{1 - A_3p - B_3q} \quad (5)$$

Equations (4) and (5) constitute a linear system with unknowns p and q, which in general has a unique solution (q.e.d).

A degenerate case in the solution of the above system arises when the centers of all three image planes are collinear. Experiments using the above method on perspective images computed the orientation of the world contour with great accuracy. Despite the fact that the paraperspective projection is an approximation of the perspective projection, and the error depends on many factors (slant, tilt, depth, size of the contour; for a detailed discussion, see [Aloimonos & Chou, 1985]), it seems that in the above method much of the introduced error is cancelled. This is a fact that was brought to our attention from extensive experiments. We are currently working towards the theoretical explanation of this error cancellation.

4. Solving the Problem with Two Frames

In the previous section, we used three frames for the recovery of shape from contour. But the information we used from the image contours was only their area, and in particular how the area was changing from view to view. A useful piece of information that we have not yet utilized is the length of the contour (which is of course independent of its area). Using this information, we can solve the shape from contour problem with two projections (binocular observer) but in a computationaliy much harder way involving nonlinear equations.

Consider a coordinate system 0,X,Y,Z, to be fixed with respect to the left camera, with the -Z axis again pointing along the optical axis. We consider that the image plane of the left camera is perpendicular to the Z axis at the point (0,0,-1). The nodal point of the right camera is the point (Δx,0,0) and the image plane of the right camera is identical to the one of the left camera (Fig. 2). C is a contour on a world plane Π with equation $-Z = pX + qY + c$, and C_l and C_R are the projections of the contour C on the left and right image respectively using the paraperspective projection. We can easily prove that a small line segment (l cos θ, l sin θ) on the image plane is due to the projection of a line segment on the world plane, with length $L = l\, L_\theta$, with

$$L_\theta = \frac{c}{(1-Ap-Bq)^2}\sqrt{k_1\cos^2\theta + k_2\sin^2\theta + k_3\sin\theta\cos\theta}$$

where:

$$k_1 = (1-qB)^2 + (pB)^2 + p^2$$

$$k_2 = (1-pA)^2 + q(A)^2 + q^2$$

$$k_3 = 2((1-qB)qA + (1-pA)\,pB + pq)\,,$$

and (A,B) is the center of gravity of the area under considertion. So, given a contour in an image, if we break the contour into small line segments (edges) ($l_i \cos\theta_i$, $l_i \sin\theta_i$), i = 1, ..., n, then the length of the contour in the world plane is given by:

$$L = \sum_{i=1}^{n} l_i L_i$$

with

$$L_i = \frac{\beta}{1-Ap-Bq}\sqrt{k_1\cos^2\theta_i + k_2\sin^2\theta_i + k_3\cos\theta_i\sin\theta_i}$$

with

$$k_1 = (1-qB)^2 + (pB)^2 + p^2$$

$$k_2 = (1\,pA)^2 + (qA)^2 + q^2$$

$$k_3 = 2((1-qB)qA + (1-pA)\,pB + pq)\,,$$

β is the depth of the center of gravity of the world contour. If we consider now the left and right images of the contour C (Fig. 3), and we compute the length of the world contour from each one, we should find the same answer. In other words, if L_L and L_R are the length of the world contour that we compute from the left and right image, respectively, we must have

$$L_L = L_R \qquad (6)$$

Equation (6) is an equation in the unknowns p,q, but it is in a complicated form that does not permit easy algebraic manipulations.

On the other hand, if S_W, S_l, S_R are the areas of the world contour, the left image contour and the right image contour respectively, then we have

$$\frac{S_L}{S_W} = \frac{1}{\beta^2}\,\frac{1-A_Lp-B_Lq}{\sqrt{1+p^2+q^2}} \qquad (7)$$

$$\frac{S_R}{S_W} = \frac{1}{\beta^2}\,\frac{1-A_Rp-B_Rq}{\sqrt{1+p^2+q^2}} \qquad (8)$$

where (A_l,B_l and (A_R,B_R) are the centers of gravity of the left and the right image contour respectively. From (7), (8), we conclude

$$\frac{S_L}{S_R} = \frac{1-A_lp-B_lq}{1-A_Rp-B_Rq} \qquad (9)$$

Equation (9) represents a straight line in gradient space, or a great circle in the (equivalent) Gaussian sphere formalism. Equations (6) and (9) constitute a nonlinear system in the unknowns p and q. We are currently working on a theoretical analysis concerning the number of the solutions of this system. Preliminary experimental

results, based on the following discrete method, indicate that there exists a unique solution. The discrete method we used is as follows: Equation (9) represents a great circle in the Gaussian sphere (constant azimuth, varying elevation). By taking different values for the elevation angle (180 values, if the different values are 1 degree apart) we solve for the gradient p,q and we choose this p,q that makes the function $(L_I - L_R)^2$ minimum.

So far, we have presented two methods for the determination of shape from contour, one based on three views and the relative change of area among the different views, and the other based on two views (binocular observer) and change of area and perimeter of the contours between the two different frames. We now proceed to a method for 3-D motion determination without having to find point to point correspondence between the successive dynamic frames.

5. Determining 3-D Motion without Correspondence

Here we only treat the case of pure translation. The general case of rotation and translation can be found in [Aloimonos & Basu 1985]. The treatment of this section presumes real perspective projection not paraperspective.

Consider a coordinate system 0XYZ fixed with respect to the camera, 0 the nodal point of the eye and the image plane perpendicular to the -Z axis, (focal length 1) that is pointing along the optical axis (Fig. 4). Let us represent points on the image plane with small letters ((x,y)) and points in the world with capital letters ((X,Y,Z)).

Let a point $P = (X_1,Y_1,Z_1)$ in the world with perspective image (x_1,y_1) where

$x_1 = X_1/Z_1$ and $y_1 = Y_1/Z_1$. If the point P moves to the position $P' = (X_2,Y_2,Z_2)$ with

$$X_2 = X_1 + \Delta X$$

$$Y_2 = Y_1 + \Delta Y$$

$$Z_2 = Z_1 + \Delta Z .$$

then we desire to find the direction of the translation $(\Delta X/\Delta Z, \Delta Y/\Delta Z)$. If the image of P' is (x_2,y_2), then the observed motion of the world point in the image plane is given by the displacement vector $(x_2 - x_1, y_2 - y_1)$ (which in the case of very small motion is also known as optic flow)

We can easily prove that

$$x_2 - x_1 = \frac{\Delta X - x_1 \Delta Z}{Z_1 + \Delta Z}$$

$$y_2 - y_1 = \frac{\Delta Y - y_1 \Delta Z}{Z_1 + \Delta Z}$$

Under the assumption that the depth is large (and the motion in depth small), the equations above become:

$$x_2 - x_1 = (\Delta X - x_1 \cdot \Delta Z)/Z \qquad (10)$$

$$y_2 - y_1 = (\Delta Y - y_1 \cdot \Delta Z)/Z . \qquad (11)$$

All the published methods for the recovery of the direction $(\Delta X/\Delta Z, \Delta Y/\Delta Z)$ are based on the above equations (10) and (11) (see [Ullman 1979, Longuet-Higgins 1981, Tsai & Huang 1984, Bandyopadhyay & Aloimonos 1985]), which of course require the knowledge of the correspondence between points in the successive frames. In the next section, we present a method for the recovery of the translational direction of a moving planar contour $(\Delta X/\Delta Z, \Delta Y/\Delta Z)$, without having to solve the correspondence problem.

6. Motion of a Planar Contour without Correspondence

Consider again a coordinate system 0XYZ fixed with respect to the camera, and a contour C on a plane Z = pX + qY + c that is moving along the vector $(\Delta X, \Delta Y, \Delta Z)$, and let C_1 and C_2 be the two successive images of the contour C (see Fig. 5). We suppose that the orientation of the contour world plane is already known (i.e. it has been found by one of the two already presented methods). In what follows, to facilitate analysis, we will present a discrete analysis (i.e. we will talk about summation over all discrete points of the contour, instead of integration along the contour).

Consider a point (x_i,y_i) on contour C_1 (the first frame of the sequence), which moves to a point (x_j,y_j) on contour C_2. For the moment we do not worry about where the point (x_j,y_j) is on the second contour C_2. The only important thing is that $(x_j,y_j) \in C_2$ and it is the corresponding point of $(x_i,y_i) \in C_1$. From that, we have

$$x_j - x_i = \frac{\Delta X - x_i \Delta Z}{Z_i} \qquad (12)$$

$$y_j - y_i = \frac{\Delta Y - y_i \Delta Z}{Z_i} \qquad (13)$$

where Z_i is the depth of the contour point whose image is the point (x_i,y_i). Taking into account that

$Z_i = pX_i + qY_i + c$ or

$1 = px_i + qy_i + c/Z_i$ or

$1/Z_i = 1/c\,(1 - px_i - qy_i)$, equations (12) and (13) become:

$$x_j - x_i = \frac{\Delta X - x_i \Delta Z}{c}(1 - px_i - qy_i) \qquad (14)$$

$$y_j - y_i = \frac{\Delta Y - y_i \Delta Z}{c}(1 - px_i - qy_i) \qquad (15)$$

Equations (14) and (15) relate the x and y coordinates respectively, of two corresponding points, the first on contour C_1 and the second on contour C_2.

If we write equation (14) for all the points on the two contours, and we sum up all these equations, we get the following

$$\sum_j x_j - \sum_i x_i = \sum_i \frac{\Delta X - x_i \Delta Z}{c} (1 - p x_i - q y_i) \qquad (16)$$

In equation (16), $\Sigma_j x_j$ denotes the sum of the x coordinates of all the points on contour C_2, $\Sigma_i x_i$ denotes the sum of the x coordinates of all the points on contour C_1 and the right hand side of the above equation is summed over all points of contour C_1. Equation (16) becomes:

$$c (\sum_j x_j - \sum_i x_i) = \Delta X \sum_i (1 - p x_i - q y_i) - \Delta Z \sum_i x_i (1 - p x_i - q y_i) \qquad (17)$$

In an analogous way, working with equation (15) we get:

$$c (\sum_j y_j - \sum_i y_i) = \Delta Y \sum_i (1 - p x_i - q y_i) - \Delta Z \sum_i y_i (1 - p x_i - q y_i) \qquad (18)$$

From equations (17), (18) we conclude:

$$\frac{\sum_j x_j - \sum_i x_i}{\sum_j y_j - \sum_i y_i} = \frac{\frac{\Delta X}{\Delta Z} \sum_i (1 - p x_i - q y_i) - \sum_i x_i (1 - p x_i - q y_i)}{\frac{\Delta Y}{\Delta Z} \sum_i (1 - p x_i - q y_i) - \sum_i y_i (1 - p x_i - q y_i)} \qquad (19)$$

Equation (19) is a linear equation on the unknowns $\Delta X/\Delta Z$ and $\Delta Y/\Delta Z$. This equation is due to the motion of the contour on one image frame. with a binocular observer, we get two linear equations (Eq. 19) from the motion of the contour in both the left and right images, which gives a unique solution for the direction of the translation ($\Delta X/\Delta Z$, $\Delta Y/\Delta Z$), without using point to point correspondence. Obviously, to apply this method in natural images, one would have to solve the problem of contour correspondence (macro-correspondence) which seems easier than point to point correspondence.

Finally, experimental results based on this method are very accurate and robust. A recent method presented by Kanatani [1985a, 1985b] has numerical instabilities that affect the desired result a great deal.

Figures 6-10 show results of binocular and trinocular experiments. Figure 6 shows the perspective images of a planar contour taken by three cameras at the positions (0,0), (0,50) and (50,0) respectively. The actual orientation of the contour in space was given by the gradient (p,q) = (15,25). The computed orientation was (p,q) = (14.99, 24.99). Figure 7 shows again the perspective images of a planar contour taken by three cameras at the positions (0,0), (0,50) and (50,0) respectively. The actual orientation of the contour in space was (p,q) = (30,5) and the estimated orientation was (p,q) = (30, 4.99). Figure 8 shows the images of a translating planar contour (human figure) taken by a binocular system at two different time instants. The actual orientation of the contour in space was (p,q) = (10,5) and the actual direction of translation (dx/dz, dy/dz) = (−4,6). Our program recovered orientation (p,q) = (10.00007, 5.000297) and direction of translation (dx/dz, dy/dz) = (−4.000309, 6.00463). Figure 9 shows again the perspective images of a translating planar contour taken by a binocular system at two different time instances. The actual orientation of the contour was (p,q) = (−25,30) and the direction of translation (dx/dz,dy/dz) = (50,60). The computed orientation from these images was (p,q) = (−24.99, 30.000021) and the computed direction of translation (dx/dz,dy/dz) = (49.858421, 59.830266). Finally, Figure 10 shows the perspective images of a translating planar contour taken by a binocular system at two different times. The actual orientation of the contour was (p,q) = (10,−11) and the direction of translation (dx/dz,dy/dz) = (1.66, 3.33). The estimated parameters from these images were (p,q) = (9.99, −11.000383) and (dx/dz,dy/dz) = (1.66, 3.33).

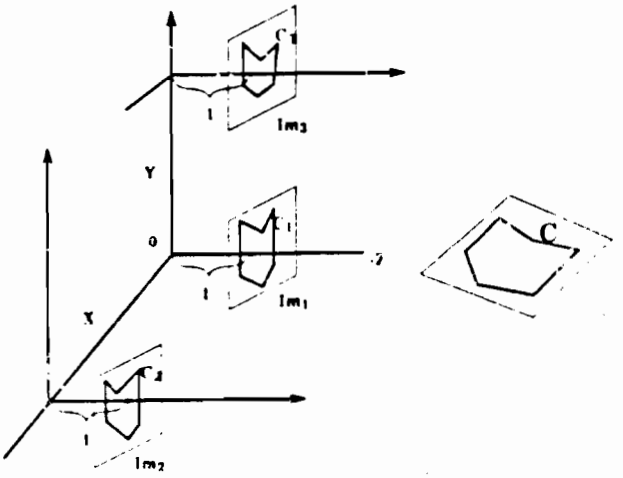


Figure 1

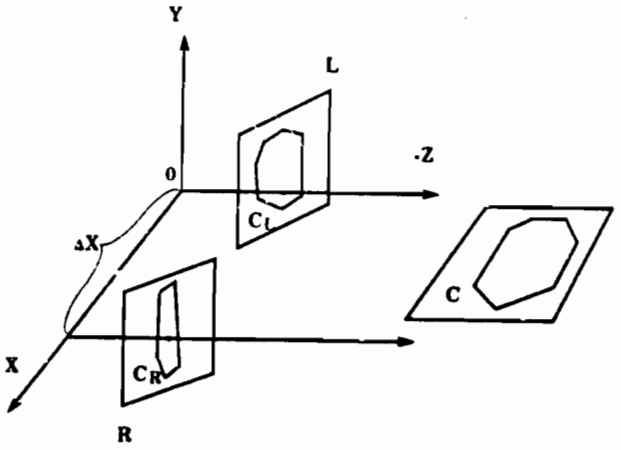


Figure 2

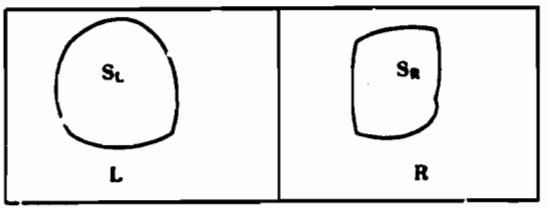


Figure 3

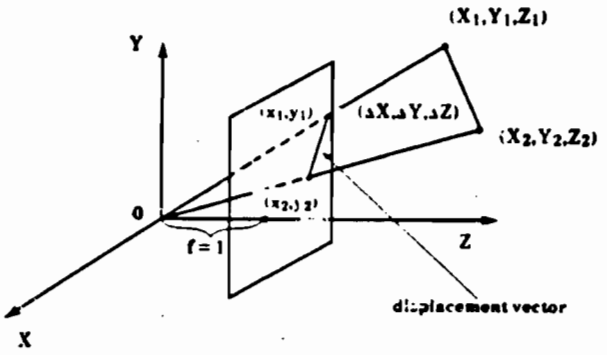


Figure 4

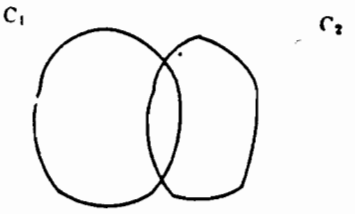


Figure 5

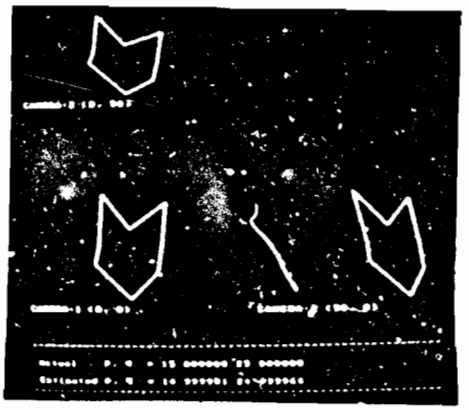


Figure 6

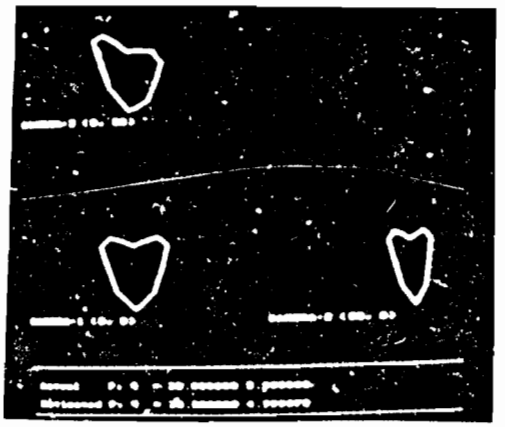


Figure 7

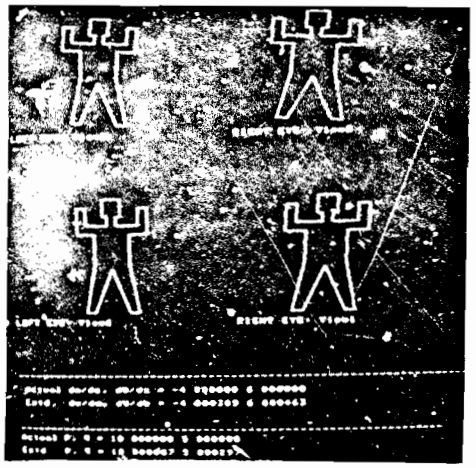


Figure 8

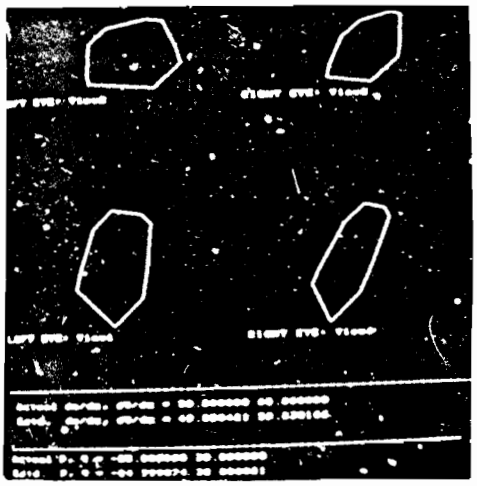


Figure 9

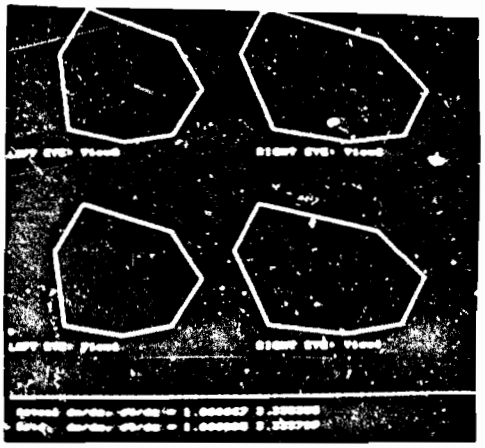


Figure 10

Acknowledgments

Our thanks go to Dana Ballard and Amit Bandyopadhyay for their help during the preparation of this paper. This research was sponsored by the Defense Advanced Research Projects Agency under Grants DACA76-85-C-0001 and N00014-82-K-0193.

References

Aloimonos, J. and Basu, A., "Shape and motion from contour," forthcoming Technical Report, Dept. of Computer Science, Univ. of Rochester, 1985.

Aloimonos, J. and Chou, P., "Detection of surface orientation and motion from texture," TR161, Dept. of Computer Science, Univ. of Rochester, January 1985.

Aloimonos, J. and M. Swain, "Shape from texture", *Proceedings*, IJCAI85, Los Angeles, CA. Aug. 1985.

Bandyopadhyay, A. and Aloimonos, J., "Perception of structure and motion of rigid objects," TR169, Dept. of Computer Science, Univ. of Rochester, 1985.

Barrow, H.G. and J.M. Tenenbaum "Recovering intrinsic scene characteristics from images", in *Computer Vision Systems*, A. Hanson and E. Riseman (eds), Academic Press, New York 1978.

Brady, M. and Yuille, A., "An extremum principle for shape from contour," *IEEE Trans. Pattern Analysis and Machine Intelligence*, **6**, 288-301, 1984.

Brown, C.M., J. Aloimonos, M. Swain, P. Chou and A. Basu "Texture, Contour, Shape and Motion," submitted to *Pattern Recognition Letters*, September 1985.

Horn, B.K.P., "Understanding image intensities," *Artificial Intelligence* **8**, (2), 201-231, 1977.

Ikeuchi, K., "Shape from regular patterns," *Artificial Intelligence* **22**, 49-75, 1984.

Ikeuchi, K. and Horn, B.K.P., "Numerical shape from shading and occluding boundaries," *Artificial Intelligence* **17**, 141-184, 1981.

Kanade, T., "Recovery of the three dimensional shape of an object from a single view," *Artificial Inteiligence* **17**, 409-460, 1981.

Kanatani, K., "Tracing planar surface motion from projection without knowing correspondence," *CVGIP*, **29**, 1-12, 1985a.

Kanatani, K., "Detecting the motion of a planar surface by line and surface integrals," *CVGIP*, **29**, 13-22, 1985b.

Kender, J.R., "Shape from texture: an aggregation transform that maps a class of textures into surface orientation," *Proceedings*, IJCAI, 475-480, 1980.

Kender, J.R., "Shape from texture: A computational paradigm," *Proceedings*, DARPA Image Understanding Workshop, April 1979, 79-84.

Longuet-Higgins, H.C., "A computer algorithm for reconstructing a scene from two projections," *Nature* **239**: 10, 133-135, 1981.

Ohta, Y., Maenobu, K., and Sakai, T., "Obtaining surface orientation of from texels under perspective projection," *Proceedings*, IJCAI, 746-751, 1981.

Tsai, R.Y. and Huang, T.S., "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **6**, 13-27, 1984.

Ullman, S., "The interpretation of visual motion," MIT Press, Cambridge, 1979.

Witkin, A., "Recovering surface shape and orientation from texture," *Artificial Intelligence* **17**, 17-45, 1981.

GC970145-19

EPIPOLAR-PLANE IMAGE ANALYSIS: A TECHNIQUE FOR ANALYZING MOTION SEQUENCES*

Robert C. Bolles
H. Harlyn Baker
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025.

Abstract

A technique for unifying spatial and temporal analysis of an image sequence taken by a camera moving in a straight line is presented. The technique is based on a "dense" sequence of images—images taken close enough together to form a solid block of data. Slices of this solid directly encode changes due to motion of the camera. These slices, which have one spatial dimension and one temporal dimension, are more structured than conventional images. This additional structure makes them easier to analyze. We present the theory behind this technique, describe an initial implementation, and discuss our preliminary results.

Introduction

Most motion-detection techniques (e.g., [Barnard 1980], [Haynes 1983], and [Hildreth 1984]) analyze pairs of images, and hence are fundamentally similar to conventional stereo techniques. A few researchers have considered sequences of three or more images (e.g., [Nevatia 1976], [Ullman 1979], and [Yen 1983]), but still the process is one of matching discrete items at discrete times. And yet, it is widely acknowledged that there is a potential benefit from unifying the analysis of spatial and temporal information. In this paper we present a technique to perform this type of unification for straight-line motions.

Motion-analysis techniques using pairs or triples of images are designed to process images that contain significant changes from one to another – features may move more than 20 pixels between views. These large changes force the techniques to tackle the difficult problem of stereo correspondence (Figure 1 shows an image triple with a typical inter-frame separation). Our idea, on the other hand, is to take a sequence of images from positions that are very close together – close enough that almost nothing changes from one image to the next. In particular, we take images close enough together that none of the image features moves more than a pixel or so (Figure 2 shows the first three images from one of our sequences containing 125 images). This sampling frequency guarantees a continuity in the temporal domain that is similar to continuity in the spatial domain. Thus, an edge of an object in one image appears temporally adjacent to (within a pixel of) its occurrence in both the preceding and following images. This temporal continuity makes it possible to construct a solid of data in which time is the third dimension and continuity is maintained over all three dimensions (see Figure 3). This solid of data is referred to as *spatio-temporal data*.

The traditional motion-analysis paradigm detects features in spatial images (i.e., the *uv* images in Figure 3), matches them from image to image, and then deduces the motion. We, however, propose an approach that is orthogonal to this. We suggest slicing the spatio-temporal data along a temporal dimension (see Figure 4), locating features in these slices, and then computing three-dimensional locations. Our reasoning is that the temporal image slices can be formed in such a way that they contain more structure than spatial images; thus, they are more predictable and, hence, easier to analyze.

To convince you of the utility of this approach, we must demonstrate that there is an interesting class of motions for which we can build structured temporal images. In the next section we show that this can be done whenever the camera moves in a straight line. We call these temporal images *epipolar-plane images*, or EPIs, from their geometric properties. In Section 3 we describe the results of our experiments in computing the depths of objects from their paths through the EPIs. And finally, in Section 4 we discuss the strengths and weaknesses of the technique and outline some current and future directions for our work.

Epipolar-Plane Images

In this section we define an epipolar-plane image (an EPI) and explain our interest in it. First, however, we review some stereo terminology. Consider Figure 5, which is a diagram of a general stereo configuration. The two cameras are modeled as pin-holes with the image planes in front of the lenses. For each point P in the scene, there is a plane, called the *epipolar plane*, which passes through the point and the line joining the two lens centers. This plane intersects the two image planes along *epipolar lines*. All the points in the epipolar plane are projected onto one epipolar line in the first image and a corresponding epipolar line in the second image. The importance of these lines for stereo processing is that they reduce the search required to find matching points from two dimensions to one. Thus, to find a match for a point along one epipolar line in an image it is only necessary to search along the corresponding epipolar line in the other image. This is termed the *epipolar constraint*.

One further definition that is essential to understanding our approach is that of an *epipole*. An epipole in a stereo configuration is the intersection of the line joining the lens centers and an image plane (see Figure 5). In motion analysis, an epipole is often referred to as a focus of expansion (FOE) because the epipolar lines radiate from it.

*This research was supported by DARPA Contracts MDA 903-83-C-0027 and DACA 76-85-C-0004

Consider a simple motion in which a camera moves from right to left, with its optical axis orthogonal to its direction of motion (see Figure 6). For this type of motion the epipolar plane for a point, such as P, is the same for all pairs of camera positions, and we refer to that plane as the epipolar plane for P for the whole motion.

The epipolar lines associated with one of these epipolar planes are horizontal scan lines in the images (see Figure 6). The projection of P onto these epipolar lines moves to the right as the camera moves to the left. The velocity of this movement along the epipolar line is a function of P's distance from the line joining the lens centers. The closer it is, the faster it moves.

For this motion, the epipolar lines are not only horizontal, they occur at the same vertical position in all the images. Therefore, a horizontal slice of the spatio-temporal data formed from this motion contains all the epipolar lines associated with one epipolar plane (see Figure 7).

Figure 7 shows three of the images used to form the solid of data. Typically a hundred or more images are used, making P's trajectory through the data a continuous path, as indicated in the diagram. For this type of lateral motion, if the camera moves a constant distance between images, the trajectories are straight lines (see Appendix A).

Figure 8 shows a horizontal slice through the solid of data shown in Figure 3, which was constructed from a sequence of 125 images taken by a camera moving from right to left. Figure 9 shows a frontal view of that slice. We call this type of image an epipolar-plane image (EPI) because it is composed of one-dimensional projections of the world points lying on an epipolar plane. Each horizontal line of the image is one of these projections. Thus, time progresses from bottom to top, and, as the camera moves to the left, the features move to the right.

There are several things to notice about this image. First, it contains only linear structures. In this respect it is much simpler than the spatial images used to create it (see Figure 1 for comparison). Second, the slopes of the lines determine the distances to the corresponding features in the world. The greater the slope, the farther the feature. Third, occlusion, which occurs when a closer feature moves in front of a more distant one, is immediately apparent in this representation. For example, the narrow white bar at the left center of the EPI in Figure 9 is initially occluded, then it is visible for a while until it is occluded briefly by a thin light object, then visible again before being rapidly occluded twice by two darker objects, and then is continuously visible until the end of the sequence. Thus, the same object is seen four different times.

Figure 10 shows another EPI sliced from the data in Figure 3. Its basic structure is the same as Figure 9; however, it illustrates the variety of patterns that can occur in an EPI.

The EPIs in Figures 9 and 10 were constructed from a simple right-to-left motion with the camera oriented at right angles to the motion. For what other types of motions can EPIs be constructed? The answer is that they can be constructed for any straight-line motion. As long as the lens center of the camera moves in a straight line the epipolar planes remain fixed relative to the scene. The points in each of these planes function as a unit. They are projected onto one line in the first image, another line in the second image, and so on. The camera can even change its orientation about its lens center as it moves along the line without affecting this partitioning of the scene. Orientation changes move the epipolar lines around in the image plane, significantly complicating the construction of the EPIs, but the epipolar planes remain unchanged since the line joining the lens centers remains fixed.

Figure 11 is an EPI formed from a sequence of images taken by a camera moving forward and looking straight ahead. Again the image is very structured, except that, instead of lines, it is composed of curves. For this type of motion, in fact for any straight-line motion in which the camera is at a fixed orientation relative to the direction of motion (see Figure 12), the trajectories in the EPI's are hyperbolas (see Appendix B). But not only are they hyperbolas, they are simple hyperbolas in the sense that their asymptotes are vertical and horizontal lines. A right-to-left motion, such as the one mentioned above, is just a special case in which the hyperbolas degenerate into lines.

If the lens center does not move in a line, the epipolar planes passing through a world point differ from one camera position to the next. The points in the scene are grouped one way for one pair of camera positions and a different way for another pair of positions. This makes it impossible to partition the scene into a fixed set of planes, which in turn means that it is not possible to construct EPIs for such a motion.

One last observation about EPIs: since an EPI contains all the information about the features in a slice of the world, the analysis of a scene can be partitioned into a set of analyses, one for each slice. In the case of a right-to-left motion, there is one analysis for each scanline in the image sequence. This ability to partition the analysis is one of the key properties of our motion-analysis technique. Slices of the spatio-temporal data can be analyzed independently (and possibly in parallel), and then the results can be combined into a three dimensional representation of the scene.

Experimental Results

We have implemented a program that computes three-dimensional locations of world features by analyzing EPIs constructed from right-to-left motions. The program currently consists of the following steps:

1. 3D smoothing of the spatio-temporal data
2. Slicing the data into EPIs
3. Detecting edges, peaks, and troughs
4. Segmenting edges into linear features
5. Merging collinear features
6. Computing x-y-z coordinates
7. Building a map of free space
8. Linking x-y-z points between EPIs

In this section we illustrate the behavior of this program by applying it to the data shown in Figure 3.

The first step smooths the three-dimensional data to reduce the effects of noise and camera jitter, and to determine the temporal contours subsequently to be used as features. This is done by applying a sequence of three one-dimensional Gaussians ([Buxton 1983] and [Buxton 1985] explore other uses of spatio-temporal convolution).

The second step forms EPIs from the spatio-temporal data. For a lateral motion this is straightforward because the EPIs are horizontal slices of the data. Figure 9 shows the EPI selected to illustrate steps three through seven.

The third step detects edge-like features in the EPI. It currently locates four types of features: positive and negative zero-crossings [Marr 1980] and peaks and troughs in the difference of Gaussians. The zero-crossings indicate places in the EPI where there is a sharp change in image intensity, typically at surface boundaries or surface markings, and the peaks/troughs occur between these zero-crossings. The former are generally more precisely positioned than the latter. Figure 13 shows all four types of features detected in the EPI shown in Figure 9.

The fourth step fits linear segments to the edges. It does this in two passes. The first pass partitions the edges at sharp corners by analyzing curvature estimates along the edges. The second pass applies Ramer's algorithm [Ramer 1972] to recursively partition the smooth segments into line segments. Figure 14 shows the line segments derived from the edges in Figure 13.

The fifth step builds a description of the line segments that links together those that are collinear. The intent is to identify sets of lines that belong to the same feature in the world. By bridging gaps caused by occlusions, the program can improve its estimates of the features' locations as well as extract clues about the nature of the surfaces in the scene. The program only links together features of the same type, except that positive and negative zero crossings may be joined, since the contrast across an edge can differ from one view to the next. Figure 15 shows the peak features from Figure 14 that are linked together by the program.

The line intersections in Figures 14 and 15 indicate temporal occlusions. For each intersection, the feature with the smaller slope is the one that occludes the other.

The sixth step computes the x-y-z locations of the world features corresponding to the EPI features. The world coordinates are uniquely determined by the location of the epipolar plane associated with the EPI and the slope and intercept of the line in the EPI. To display these three-dimensional locations, the program plots the two-dimensional coordinates of the features in the epipolar plane. Figure 16 shows the epipolar plane coordinates for the features shown in Figure 14. The shape and size of each ellipse depicts the error associated with the feature's location (this depends on the length of the line and the variance of the fit).

The seventh step builds a two-dimensional map of the world that indicates which areas are empty (also see [Bridwell 1983]). This construction is demonstrated for a few points in Figure 17, where the Z = 0 axis is the camera path. The principle here is that if a feature is seen continuously over some interval by a moving camera, then during that motion nothing occludes it. Since nothing occludes it, nothing lies in front of it, and the triangle in the scene defined by the feature and its first and last points of observation is empty space. We build a map of this free space by constructing one of these triangular regions for each line segment found in the EPI, and then ORing these all together. Notice in Figure 17 that one of the features is viewed once while the other is seen in two distinct intervals, and so gives rise to two free-space triangles. Figure 18 shows the full free-space map constructed for the EPI features of Figure 16. To take the INTERSECTION of the free-space maps from these individual EPIs, perhaps over some vertical interval, would give us the known free-space volume in that interval. This would be useful for navigation, as we know we could move freely in that volume without running into obstacles.

Figures 19 through 22 show the processing for the EPI 30 lines from the bottom of the uv images. This slice contains a plant on the left, a shirt draped over a chair, part of the top of a table, and in the right foreground, a ladder.

Figure 23 is a stereo (crossed-eye) display, showing some preliminary results in the eighth step of our analysis - combining the spatial data from the individual EPIs. For spatial continuity, we link points between the various EPIs (nearest neighbors in overlapping error ellipses). Figure 23 displays those features whose total baseline is greater than 3 inches, and whose connected extent vertically is greater than 2 scanlines.

As a final depiction of our results in this depth from camera motion analysis, we show in Figure 24 all mapped features visible from the 90th frame of the sequence. Figure 25 shows these features separated into the specific depth ranges indicated.

Discussion

The following valuable characteristics of this approach should be noted:

- Spatial and temporal data are treated together as a single unit;
- The acquisition and tracking steps of the conventional motion analysis paradigm are merged into one step;
- The approach is feature-based, but is not restricted to point features – linear features that are perpendicular to the direction of motion can also be used;
- There is more structure in an EPI than in a standard spatial image, which means that it is easier to analyse, and hence easier to interpret;
- Occlusion is manifested in an EPI in a way that increases the chance of detection because the edge is viewed over time against a variety of backgrounds;
- EPIs facilitate the segmentation of a scene into opaque objects occurring at different depths because they encode a *homogeneous* slice of the object over time;
- There are some obvious ways to make the analysis incremental in time and partitionable in y (epipolar planes), for high speed performance.

With these benefits, the inherent limitations and current restrictions must be borne in mind:

- Motion must be in a straight line and (currently) the camera must be at a fixed angle relative to the direction of motion;
- Frame rate must be high enough to limit the frame-to-frame changes to a pixel or so (more specifically, such that the projective width of a surface is greater than its motion);
- Independently moving objects will either not be detected, or will be detected inaccurately.

We are currently investigating the following areas:

- Extending our analysis of connectivity between adjacent EPIs - this seems to be best handled by not losing the information in the first place, that is, by making explicit the feature connectivity in space as well as time.
- Identifying and interpreting spatial and temporal phenomena such as occlusions, shadows, mirrors, and highlights.
- Characterizing the appearance of curved surfaces in EPIs.
- Implementing the analysis of EPIs derived from forward motions.
- Providing more dense depth information by, for example, tracking intensity levels.
- Making the analysis incremental in T, rather than V - that is, processing spatial images over time, rather than requiring the acquisition of all images, and then processing by EPI slices.

Appendix A: Lateral-Motion Trajectories

In this appendix we first derive an equation for the trajectory of a point in an EPI constructed from a lateral motion, and then show how to compute the (x,y,z) location of such a point. Figure 26 is a diagram of a trajectory in an EPI derived from the right-to-left motion illustrated in Figure 27. The scanline at t_1 in Figure 26 corresponds to the epipolar line l_1 in Figure 27. Similarly, the scanline at t_2 corresponds to the epipolar line l_2. (Recall that the EPI is constructed by extracting one line from each image taken by the camera as it moves along the line joining c_1 and c_2. Since the images are taken very close together in time, there would be several images taken between c_1 and c_2. However, to simplify the diagram none of these is shown.) The point (u_1, t_1) in the EPI corresponds to the point (u_1, v_1) in the image taken by the camera at time t_1 and position c_1. Thus, as the camera moves from c_1 to c_2 in the time interval t_1 to t_2, the scene point *moves* in the EPI from (u_1, t_1) to (u_2, t_2). The intent of this section is to characterize the shape of this trajectory and then compute the three-dimensional position of the corresponding scene point, given the focal length of the camera, the camera speed, and the coordinates of points along the trajectory.

For our analysis we define a left-handed coordinate system that is centered on the initial position of the camera (i.e., c_1 in Figure 27). The shape of the trajectory can be derived by analyzing the geometric relationships in the epipolar plane that passes through P. Figure 28 is a diagram of that plane.

Given the speed of the camera, s, which is assumed to be constant, the distance from c_1 to c_2, Δx, can be computed as follows:

$$\Delta x = s\,\Delta t \tag{1}$$

where Δt is $(t_2 - t_1)$. By similar triangles

$$\frac{u_1}{h} = \frac{x}{D} \tag{2}$$

$$\frac{u_2}{h} = \frac{\Delta x + x}{D} \tag{3}$$

where u_1 and u_2 have been converted from pixel values into distances on the image plane, h is the distance from the lens center to the epipolar line in the image plane, x is the x-coordinate of P in the scene coordinate system, and D is the distance from P to the line joining the lens centers. Since h is the hypotenuse of a right triangle, it can be computed as follows:

$$h = \sqrt{f^2 + v_1^2}, \tag{4}$$

where f is the focal length of the camera. From 2 and 3 we get

$$\Delta u = (u_2 - u_1) = \frac{h(\Delta x + x)}{D} - \frac{h\,x}{D} = \frac{h}{D}\Delta x \tag{5}$$

Thus, Δu is a linear function of Δx. Since Δt is also a linear function of Δx, Δt is linearly related to Δu, which means that trajectories in an EPI derived from a lateral motion are straight lines.

The (x, y, z) position of P can be computed by scaling u_1, v_1, and f appropriately. From 5 we define

$$m = \frac{D}{h} = \frac{\Delta x}{\Delta u} \tag{6}$$

which represents the slope of the trajectory computed in terms of the distance traveled by the camera (Δx as opposed to Δt) and the distance the point moved along the epipolar line (i.e., Δu). From similar triangles

$$(x, y, z) = \left(\frac{D}{h}u_1, \frac{D}{h}v_1, \frac{D}{h}f\right) \tag{7}$$

which means that

$$(x, y, z) = (mu_1, mv_1, mf) \tag{8}$$

If the first camera position, c_1, on an observed trajectory is different from the camera position, c_0, that defines a global camera coordinate system, the x coordinate can be adjusted by an amount equal to the distance traveled from c_0 to c_1. Thus,

$$(x, y, z) = ((t_1 - t_0)s + mu_1, mv_1, mf) \tag{9}$$

where t_0 is the time of the first image and s is the speed of the camera. This correction is equivalent to computing the x intercept of the line and using it as the first camera position. Therefore, for a lateral motion, the trajectories are linear and the (x, y, z) coordinates of the points can be easily computed from the slopes and intercepts of the lines.

Appendix B: Forward-Motion Trajectories

The derivation of the form of a trajectory produced by a forward motion is similar to the one used for lateral motion. Figure 29 is a diagram of a trajectory in an EPI derived from a sequence of images taken by a camera moving in a straight line at a fixed orientation relative to the principal axis of the camera (see Figure 30). Without loss of generality we have rotated the image plane coordinate systems in a uniform way so that the epipoles are on the u axes. The EPI in Figure 29 was constructed by extracting pixel intensities along epipolar lines in the images shown in Figure 30 and inserting them as scanlines in Figure 29. For example, epipolar line l_1 was placed at t_1, l_2 was placed at t_2, and so on. The point (u_1, t_1) in the EPI corresponds to the point (u_1, v_1) in the image taken at time t_1 and position c_1. Thus, as the camera moves from c_1 to c_2 over the time interval t_1 to t_2, the scene point *moves* in the EPI from (u_1, t_1)

to (w_2, t_2). Our goal is to characterize the shape of this trajectory, and then compute the three-dimensional position of the corresponding scene point, given the focal length of the camera, the camera speed, the angle between the camera's axis and the direction of motion (θ), and the coordinates of points along the trajectory.

As before, we define a left-handed coordinate system that is centered on the initial position of the camera (i.e., c_1 in Figure 30). The shape of the trajectory can be derived by examining the geometric relationships in the epipolar plane that passes through P. Figure 31 is a diagram of that plane.

Given the speed of the camera, s, which is assumed to be constant, the distance from c_1 to c_2, Δe, can be computed as follows:

$$\Delta e = s\,\Delta t \tag{10}$$

where Δt is $(t_2 - t_1)$. By similar triangles

$$\frac{w_1}{h} = \frac{C}{e} \tag{11}$$

and

$$\frac{w_2}{h} = \frac{C}{(e - \Delta e)} \tag{12}$$

where w_1 and w_2 are distances on the image plane, h is the distance from the lens center to the epipole, C is the distance from P to the line joining the lens centers measured in a plane parallel to the image planes, and e is the distance along the line joining the lens centers from c_1 to the plane passing through P and parallel to the image planes. Since h is the hypotenuse of a right triangle (see Figure 30), it can be computed as follows:

$$h = \frac{f}{\cos\theta} \tag{13}$$

where f is the focal length of the camera. From 11 and 12 we get

$$\Delta w = (w_2 - w_1) = \frac{hC}{(e - \Delta e)} - \frac{hC}{e} = \frac{hC\Delta e}{e(e - \Delta e)} \tag{14}$$

which can be rewritten as

$$e\,\Delta w\,\Delta e - e^2\Delta w + hC\Delta e = 0. \tag{15}$$

Using 10 to express Δe in terms of Δt, this becomes

$$se\Delta w\Delta t - e^2\Delta w + shC\Delta t = 0. \tag{16}$$

which defines a hyperbola whose asymptotes are the lines $w = 0$ and $t = e/s$ (see Figure 32). Thus, the trajectory is a hyperbola in which the point P appears arbitrarily close to the epipole when the camera is far away from it (as one would expect), and the projection of P moves away from the epipole at an increasing rate as the camera gets closer to it. This relationship agrees intuitively with the fact that a projective transformation involves a $1/z$ factor, which makes u a hyperbolic function of z.

Equation 14 can be used to compute z. First, rewrite it as follows:

$$\Delta w = \frac{hC}{(e - \Delta e)}\frac{1}{e}\Delta e \tag{17}$$

Then using Equation 12 and

$$e = \frac{z}{\cos\theta} \tag{18}$$

we get

$$\Delta w = \frac{w_2\cos\theta\,\Delta e}{z} \tag{19}$$

or

$$z = \frac{w_2\cos\theta\, s\Delta t}{\Delta w} \tag{20}$$

Notice that it is NOT necessary to determine the coefficients of the hyperbola in order to compute z. Two points on the trajectory are sufficient to compute Δt and Δw, which in turn, are sufficient to compute z. Also notice, however, that it is easy to fit an hyperbola of this type because it is in the simple form

$$\Delta w\Delta t + a\Delta w + b\Delta t = 0, \tag{21}$$

which is linear with respect to the coefficients a and b. This type of fitting provides a way to increase the precision with which the scene points are located.

The expression for z in Equation 20 does not apply when $\theta = 90°$, but that is the lateral motion case covered earlier. Thus, the trajectories are always hyperbolas; they just happen to degenerate into straight lines when $\theta = 90°$, which corresponds to the case in which the epipoles are not in the image plane, but rather lie at infinity.

The x and y coordinates for P can be computed by scaling z appropriately:

$$(x, y) = \left(\frac{u_1}{f}z, \frac{v_1}{f}z\right) \tag{22}$$

Recall that u_1 and v_1 are measured in a rotated-image-plane coordinate system that was set up to place the epipole on the u axis. Therefore, in addition to converting pixel values to a standard metric, such as meters, the image coordinates of a point must be rotated about the principal axis before they can be inserted into Equation 22. To compute a world-centered position for P, the (x, y, z) position computed by Equations 20 and 22 has to be transformed for the initial position of the camera along the path.

References

[Barnard 1980] "Disparity Analysis of Images," S. T Barnard and W. B. Thompson, *IEEE Trans., PAMI*, Vol 2, No 4, 1980.

[Bridwell 1983] "A Discrete Spatial Representation for Lateral Motion Stereo," N. J. Bridwell and T. S. Huang, *Computer Vision, Graphics, and Image Processing*, Vol 21, 1983.

[Buxton 1983] "Monocular Depth Perception from Optical Flow by Space Time Signal Processing," B. F. and Hilary Buxton, *Proceedings of the Royal Society of London, B.*, Vol 218, 1983.

[Buxton 1985] "Computation of optic flow from the motion of edge features in image sequences," B. F. and H. Buxton, *Image and Vision Computing*, Vol 2, No 2, May 1985.

[Haynes 1983] "Detection of Moving Edges," S. M. Haynes and R. Jain, *Computer Vision, Graphics, and Image Processing*, Vol 21, No 3, 1983.

[Hildreth 1984] "Computations Underlying the Measurement of Visual Motion," E. C. Hildreth, *Artificial Intelligence*, Vol 23, 1984.

[Marr 1980] "Theory of Edge Detection," D. C. Marr and E. Hildreth, *Proceedings of the Royal Society of London*, B 207, 1980.

[Nevatia 1976] "Depth Measurement from Motion Stereo," Ramakant Nevatia, *Computer Graphics and Image Processing*, 5, 1976.

[Ramer 1972] "An Iterative Procedure for the Polygonal Approximation of Plane Curves " U. Ramer, *Computer Graphics and Image Processing* Vol 1, 1972.

[Ullman 1979] *The Interpretation of Visual Motion*, S. Ullman, MIT Press, Cambridge, Mass., 1979.

[Yen 1983] "Determining 3-D Motion and Structure of a Rigid Body Using the Spherical Projection," B. L. Yen and T. S. Huang, *Computer Graphics and Image Processing*, Vol 21, 1983.

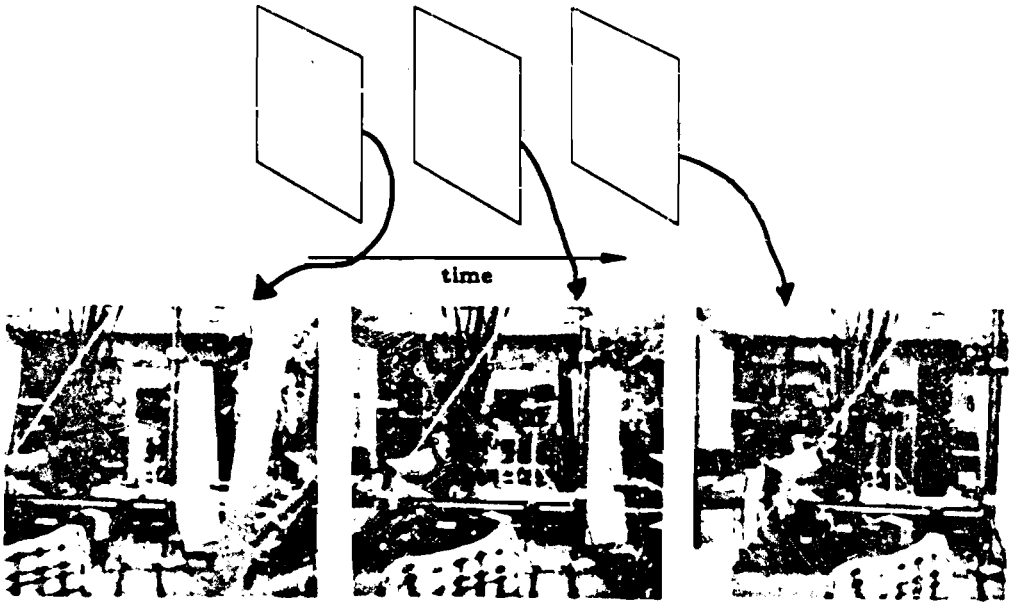


Fig. 1. Typical image separation

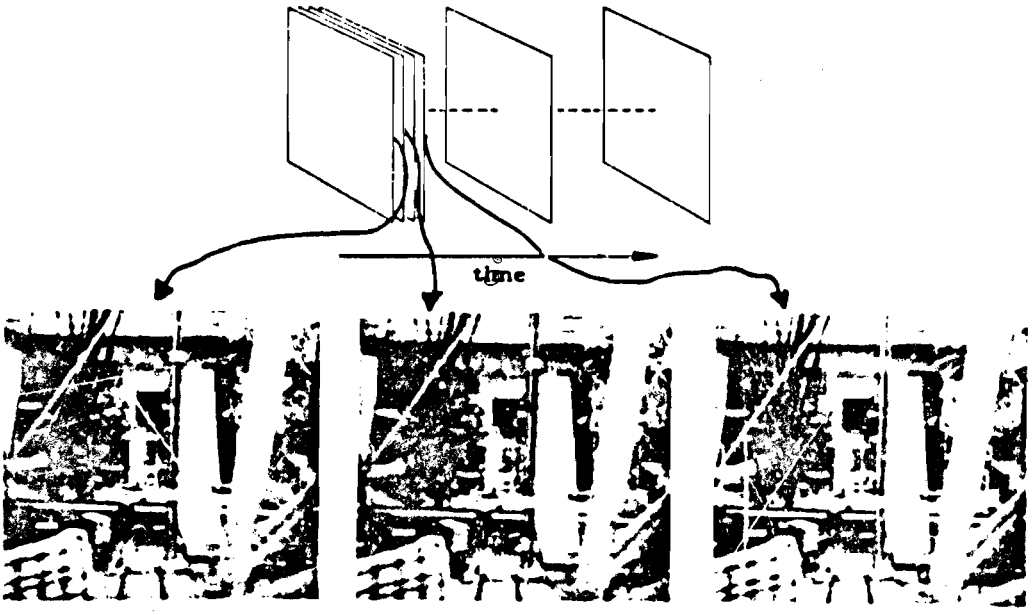


Fig. 2. Close sampling image separation

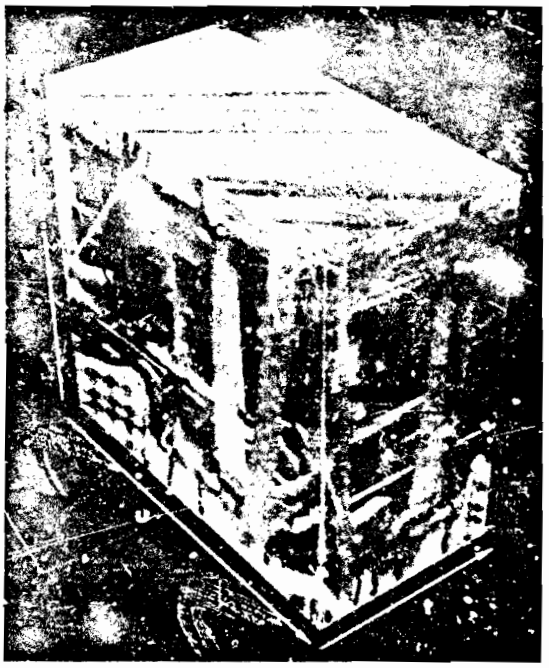


Fig. 3. Spatio-temporal solid of data

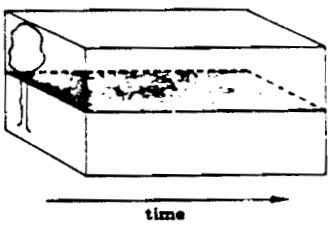


Fig. 4. Slice of the solid of data

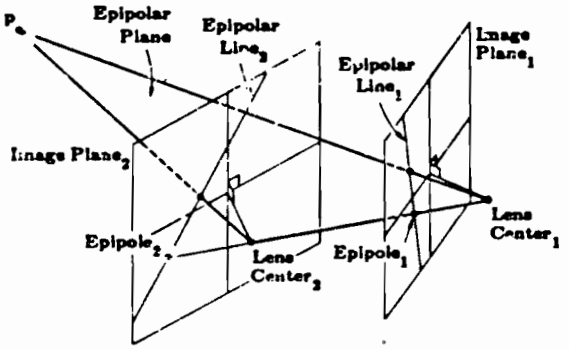


Fig. 5. General stereo configuration

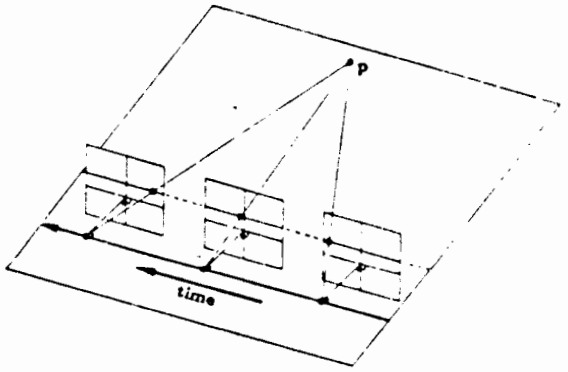


Fig. 6. Right-to-left motion

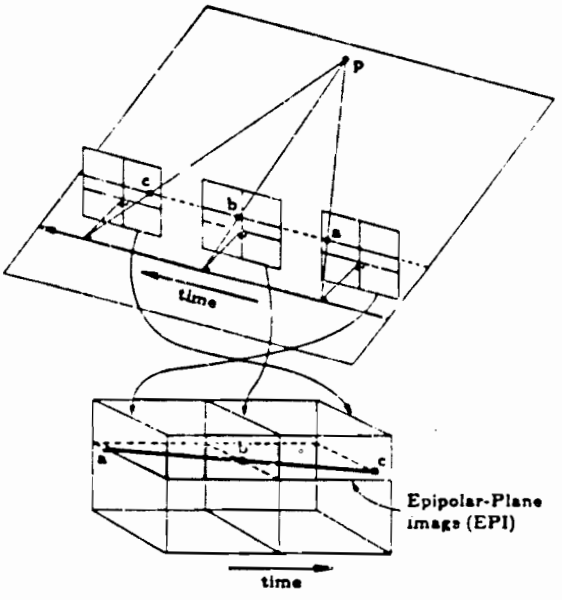


Fig. 7. Sliced solid of data

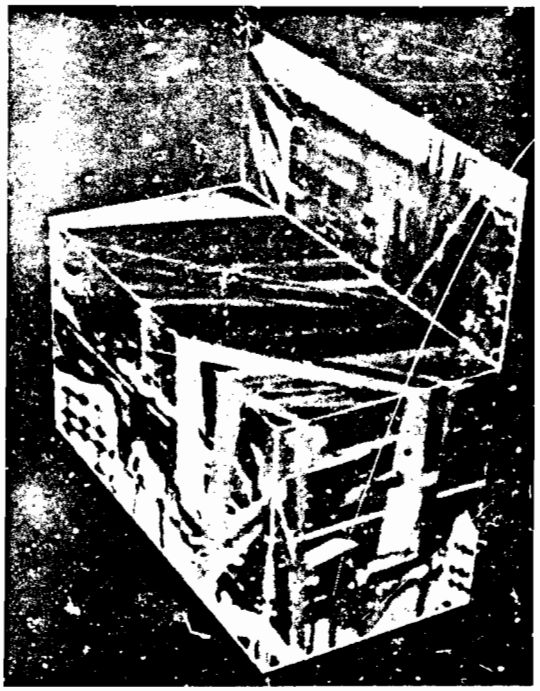


Fig. 8. Right-to-left motion with solid

Fig. 11. EPI from forward motion

Fig. 9. Frontal view of the EPI

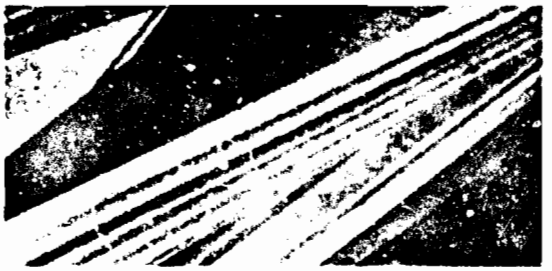


Fig. 10. A second EPI

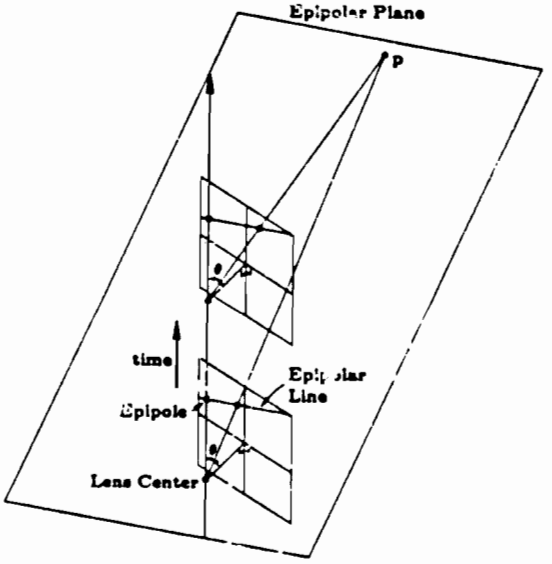


Fig. 12. Forward motion

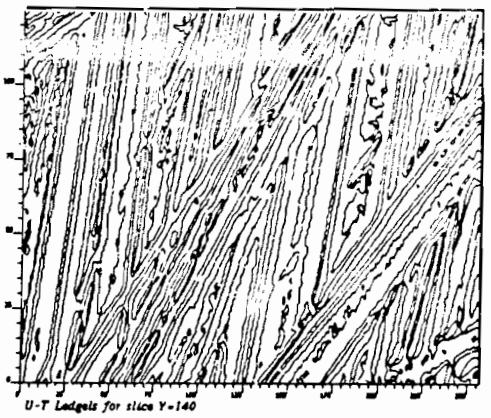


Fig. 13. Edge features in EPI

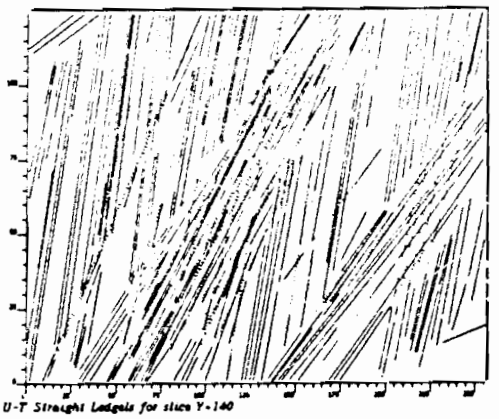


Fig. 14. Straight lines

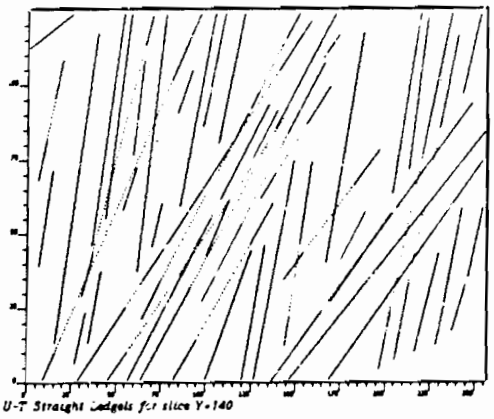


Fig. 15. Linked peak lines

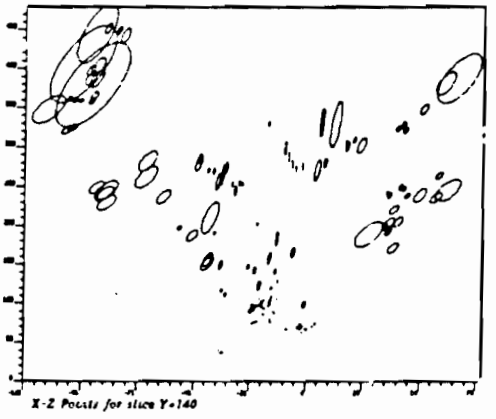


Fig. 16. xz locations

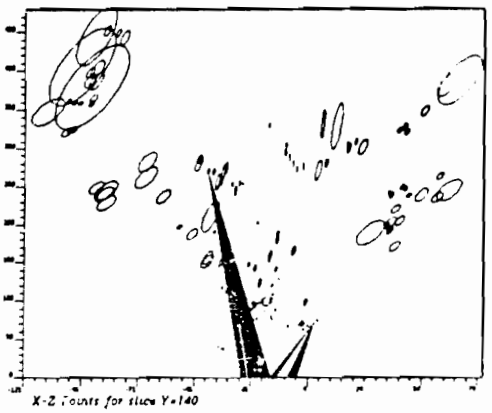


Fig. 17. Free space for 2 features

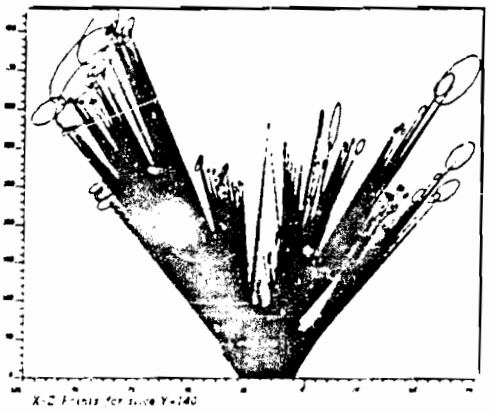


Fig. 18. Free space

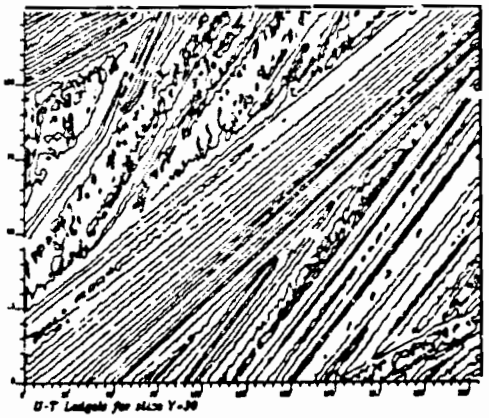


Fig. 19. Edge features in EPI

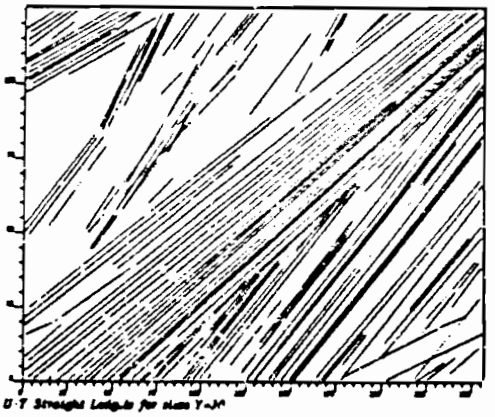


Fig. 20. Straight lines

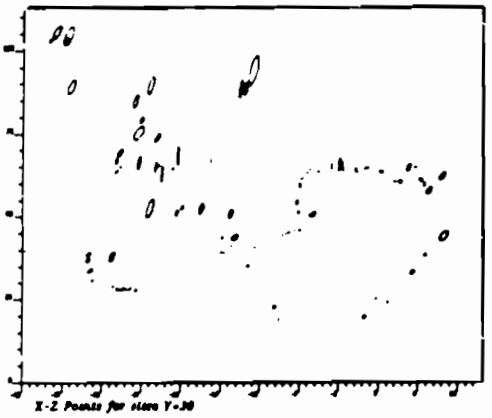


Fig. 21. xs locations

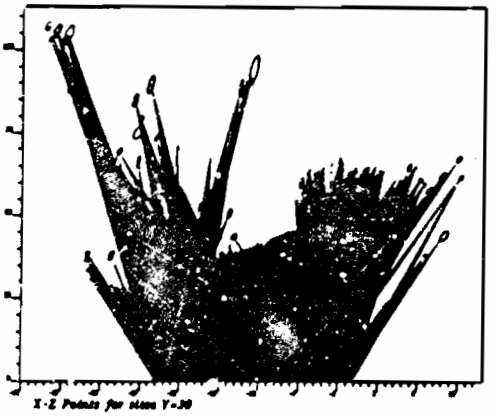


Fig. 22. Free space

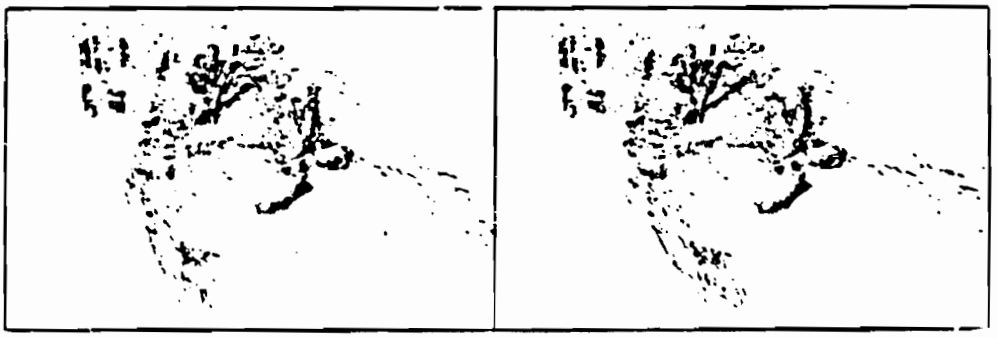


Fig. 23. Display of linked x-y-z points

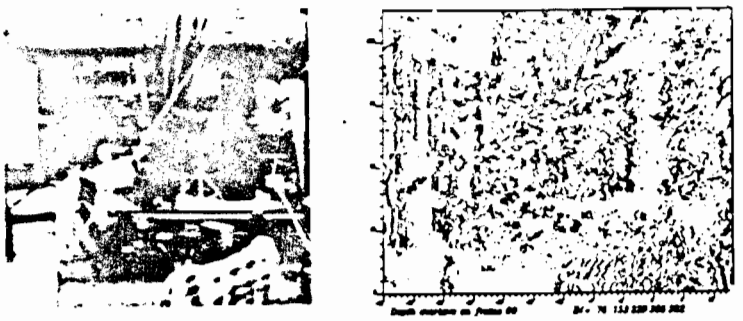


Fig. 24. Frame 90 and matched features

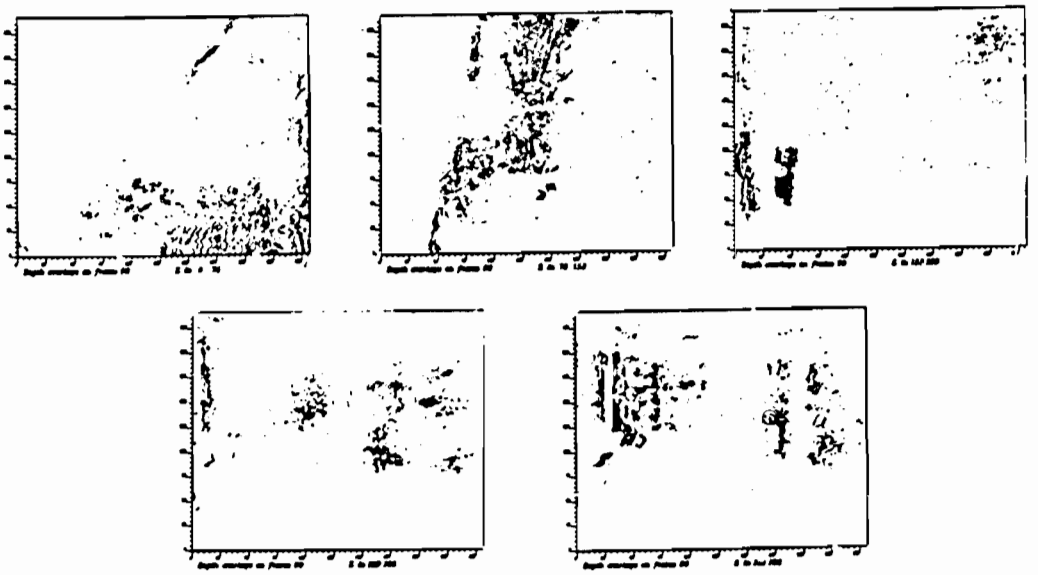


Fig. 25. Matched features over five depth intervals

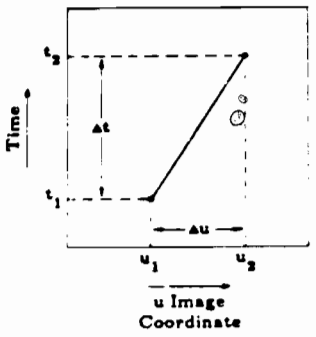


Fig. 26. Lateral motion EPI

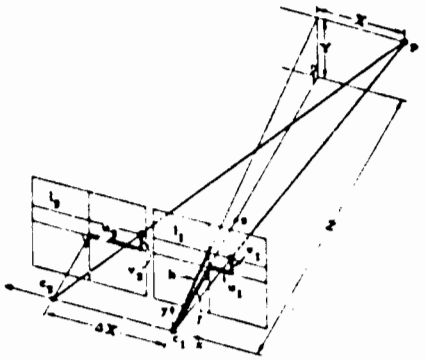


Fig. 27. Lateral motion geometry

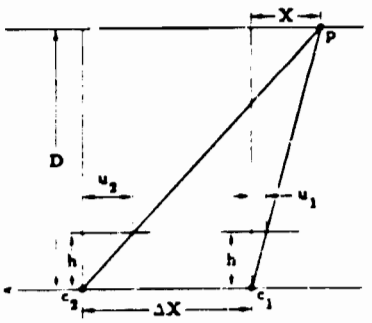


Fig. 28. Lateral motion epipolar geometry

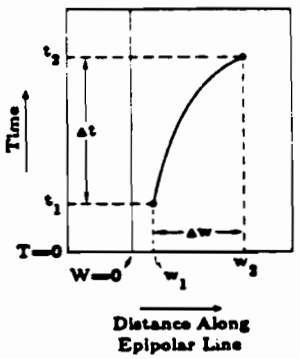


Fig. 29. Forward motion EPI

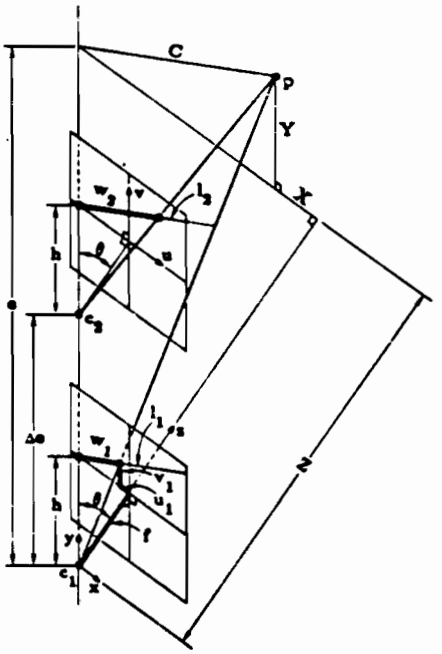


Fig. 30. Forward motion geometry

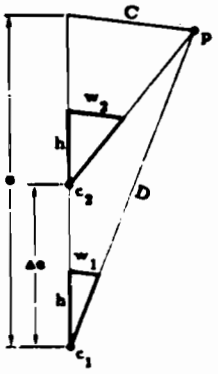


Fig. 31. Forward motion epipolar plane

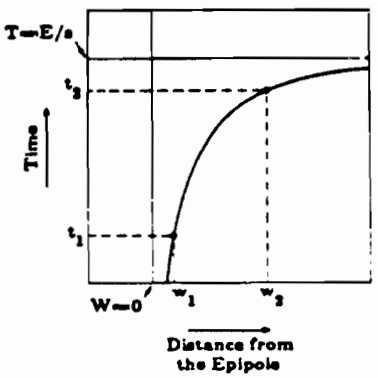


Fig. 32. Asymptotes for the hyperbola

GC970145-20

SRI's Baseline Stereo System

Marsha Jo Hannah

Artificial Intelligence Center, SRI International
333 Ravenswood Ave, Menlo Park, CA 94025

Abstract

We have implemented a baseline system for automated, area-based stereo compilation. This system, STEREOSYS, operates in several passes over the data, during which it iteratively builds, checks, and refines its model of the 3-dimensional world, as represented by a pair of images. In this paper, we describe the components of STEREOSYS and give examples of the results it produces. We find that these results agree quite well with the best available benchmark—results produced on the interactive DIMP system at the U.S. Army Engineer Topographic Laboratories.

1 Introduction

Automatic techniques for the production of 3-dimensional data via stereo compilation are receiving increased interest for a variety of applications, including cartography [Panton, 1978], autonomous vehicle navigation [Hannah, 1980], and industrial automation [Nishihara and Poggio, 1983]. Conventional stereo compilation techniques, which are based on area correlation, can produce incorrect results under a variety of conditions, for example, when views are widely separated in space or time, in the vicinity of partial occlusions, in featureless or noisy areas, and in the presence of repeated patterns.

We are investigating ways to overcome these inadequacies. Our research strategy is first to implement a baseline system that performs conventional stereo compilation, then to replace pieces of the system with improved modules as we develop them. Thus, our baseline system forms the core of an ever-improving stereo system. We have also tested the baseline system [Hannah, 1985-a] against a "challenge data base" [Hannah, 1985-b] of image areas where conventional stereo techniques encounter difficulty.

As currently implemented, our system includes routines to perform the following operations automatically:

- Construct hierarchies for stereo images
- Select "interesting" points for sparse matching
- Search 2D regions for sparse matches
- If necessary for uncalibrated imagery, compute relative camera parameters from sparse matches
- Compute epipolar lines
- Locate epipolar matches, using disparity estimates from sparse matches when available
- Evaluate matched points for believability
- Interpolate between matched points
- Display images and results in left-right stereo, red-green stereo, or as a monocular disparity field
- Compute range data and x-y-z coordinates for matched point pairs
- Display terrain data in perspective with hidden lines removed.

We are currently exploring improved techniques for image matching and match evaluation.

2 The Stereo System

SRI has integrated existing pieces of stereo code into a baseline system for automated area-based stereo compilation, then improved the system to its present form. The system operates in several passes over the data, during which it iteratively builds, checks, and refines its model of the 3-dimensional world represented by a pair of images.

The driving program is called STEREOSYS (STEREO SYStem). It allows the user to invoke a variety of modules to perform the necessary processing for stereo compilation. In theory, the modules are independent and can be replaced with improved versions at will; in practice, there are some unavoidable interdependencies of global variables that will have to be attended to.

The following sections describe the components of STEREOSYS in the order they are normally invoked; examples of their results are included. Comments are also made as to improvements that could be made to each of the modules.

2.1 Preliminary Processing

Before the actual stereo matching can begin, some preliminary image processing is necessary. This includes the creation of the image hierarchy and the selection of the interesting points to be matched.

2.1.1 Creating the Image Hierarchy

The basis for the image matching techniques is a hierarchy of images, as shown in Figure 1. The module of STEREOSYS that forms this hierarchy from the original images is called REDUCE. In the example used for the figures, the original images are a pair of image "chips" digitized from standard 9" × 9" mapping pho-

tos taken over Phoenix South Mountain Park, near Guadalupe (a suburb of Phoenix), Arizona. These images are 2048 × 2048 pixels in size, and cover an area that is approximately 2 kilometers square on the ground; elevations in the area range from 360 to 540 meters. The reduction hierarchy consists of a pyramid of images, each at half of the resolution of its parent; in this case REDUCE produces pairs of images that are 1024 × 1024, 512 × 512, 256 × 256, 128 × 128, 64 × 64, 32 × 32, and 16 × 16 pixels in size. (Figure 1 shows only the 256 × 256 through 16 × 16 image pairs.)

REDUCE ordinarily produces pixels in each reduced image by convolving the image with a Gaussian, then sub-sampling [Burt, 1981]. Older code also exists to reduce images by simple averaging of the pixels in an N × N square from the next-largest image (in most cases, N=2). It is known that this technique can produce artifacts in the data, and the more sophisticated Gaussian technique is preferred.

2.1.2 Selecting Interesting Points

The first step in the matching process is to procure a set of well-scattered, reliable matches in the image. Our approach is first to select areas in one image that contain sufficient information to produce reliable matches. To accomplish this, a statistical operator based on image variance and edge strength is passed over the image; local peaks in the output of this operator are recorded as the preferred places to attempt the matching process.

Historically, such operators have been called *interest* operators, and the peaks in the operator output have been called *interesting points* [Moravec, 1980]. This nomenclature is somewhat misleading, as the points selected are rarely interesting to a human observer; however, these terms have been in use in the computer vision community for over 10 years. It should be noted that present interest operators are not feature detectors; the same operator run over both images of a stereo pair will not necessarily pick out the same points in the two images. In our system, the interest operator is run in only one of the images, where it selects points that are to be matched in the second image by various correlation techniques. (A possible enhancement to STEREOSYS would be to design and implement efficient interest operators that really do choose "interesting" points, such as crossroads, building corners, sharp bends in rivers, etc.)

The module INTEREST permits the user to specify the interest operator to be used [Hannah, 1980], the window size over which it is calculated, and the minimum spacing for interesting points. It also provides the capability to divide the image into a grid of subimages, and records the relative ranks of the interesting points within their grid cells; this permits the most interesting point(s) in each area to be matched first. Figure 2 shows the interesting points for the right image of the Phoenix pair; the numbers indicate the 1st, 2nd, 3rd, and 4th most interesting points in a 6 × 6 grid of cells.

2.2 Preliminary Matching

At this point in the processing, it is possible to take one of two different approaches to the matching. If nothing is known regarding the absolute camera positions and orientations (as would be the case for a stereo pair taken with handheld cameras), an unstructured hierarchical matching algorithm is used on the most interesting points. The results of these matches are used in seeking a solution for a simplistic relative camera model (5 angles describing the relative positions and orientations of 2 ideal pinhole cameras [Hannah, 1974]), which can then be used for the epipolar constraint in further matching. On the other hand, if the camera parameters are known (as would be the case for the highly calibrated cartographic stereo images intended for terrain mapping) matching can proceed directly with the epipolar constraints.

2.2.1 Unconstrained Hierarchical Matching

Unconstrained hierarchical matching is done by the module HMATCH. HMATCH assumes that nothing is known about the relative orientations of the images, other than that they cover approximately the same area, at about the same scale, with no major rotation between the images. It matches each specified point (usually the most interesting point in each grid cell) using an unguided hierarchical matching technique similar to that reported in Moravec [1980]. This technique begins with the point in the largest image (the 2048 × 2048 right image of the Phoenix set), traces it back through that image's hierarchy (in our example, it repeatedly halves the co-ordinates of the point) until it reaches an image that is approximately the size of the correlation window (the 16 × 16 image for the 11 × 11 correlation windows that we used). It then uses a 2-dimensional spiral search, followed by a hill-climbing search for the maximum of the normalized cross-correlation between the image windows [Quam, 1971]. This global match is then refined back down the image hierarchy; that is, the disparity at each level (suitably magnified to account for relative image scales) is used as a starting point for a hill-climb search at the next level. The plausibility of the final match is then checked by reversing the roles of the right and left images and repeating the unconstrained hierarchical search, starting with the just-found matching point. In order for the match to be believed, this reverse search must produce a match at (or immediately adjacent to) the original interesting point. The addition of this constraint, which effectively requires co-operative results between the two images, has greatly reduced the number of bad matches in photographs which violate one or more of the assumptions around which STEREOSYS was designed. The correlation window size remains constant at all levels of the hierarchy, so the match is effectively performed first over the entire image, then over increasingly local areas of the image. This technique permits the use of the overall image structure to set the context for a match; the gradually increasing detail in the imagery is then followed down through the hierarchy to the final match.

Figure 3 shows the results of this technique on a point in the Phoenix set. The image hierarchy is the same as in Figure 1, with the addition of image chips covering the matched area in the 2048 × 2048, 1024 × 1024, and 512 × 512 images; these are shown in the upper right corner of each hierarchy. The matching began in the right image in the 2048 × 2048 chip, traced this point through the right-hand hierarchy (approximately clockwise in the figure) to the 16 × 16 right image, matched that to the 16 × 16 left image, then refined the match back through the left image hierarchy until reaching the left 2048 × 2048 chip.

It is instructive to look at the correlation coefficients for these matches (see Table 1). In the smaller images, the correlation is poor, since the window covers a large area of terrain with a great deal of relief. As the matching moves up the hierarchy, the correlation improves, because the window now approximates an area

at a single elevation. After reaching the 256 × 256 images, however, the correlation begins to decline, both in absolute value and with respect to an autocorrelation-based threshold [Hannah, 1974]. This is due to noise in the images; if one examines the chip from the 2048 × 2048 left image, one will see several streaks across the image, representing scratches on the original photograph and/or dropped data in the digitization; close examination also reveals a grainy noise pattern. Because the degraded correlations will cause difficulties in determining which matches are the correct ones, our processing has gone only to the 1024 × 1024 images, the highest resolution image in which the noise was considered tractable. Once processing is complete, STEREOSYS can be used to refine the final matches from this level down to the original 2048 × 2048 images.

Figure 4 shows the results of HMATCH on the most interesting point in each grid cell. Only the points thought to have been matched correctly are shown; those with poor correlation or whose matches fell outside of the image have been discarded by STEREOSYS.

2.2.2 Relative Camera Model Calibration

If no camera calibration information is available, the module C2MODEL can be used to calculate a simplistic relative camera model from a set of matched point pairs. This is accomplished by searching for 5 angles—the azimuth and elevation of the second camera's focal point with respect to the first camera; and pan, tilt, and roll of the second camera's axes with respect to those of the first. The object of the search is to minimize the error between the matched point in the second image and the epipolar line produced when the point in the first image is projected through the hypothesized pinhole cameras. The search proceeds by a linearization of the equations and their analytic derivatives [Gennery, 1980]. Once a solution is found, the reliability of the matched points is assessed. Points that appear to contribute too much error to the solution are removed from the calculation, and the solution is redone. Either this process reaches a successful conclusion when the point set is found to be consistent, or it reports failure if too many of the point pairs are rejected.

The resulting camera model is quite crude, as it must depend on a guess as to the focal lengths of the cameras and the length of the baseline between the cameras. Also, it assumes that we are using pinhole cameras, thus totally ignoring the internal geometry of real cameras. However, in many cases, it is suitable for approximating the epipolar constraint to simplify further matching.

2.2.3 Epipolar Constrained Hierarchical Matching

If the camera parameters are given (or once the crude ones have been derived), matching can proceed somewhat more efficiently. The camera parameters define the manner in which a point in the first image projects to a line in the second image—the epipolar constraint. This constraint can be used to cut the search from two dimensions (all over the image) to one dimension (back and forth along the epipolar line), as implemented in the module LMATCH.

LMATCH proceeds very much like HMATCH, except that the search for a match is confined to the vicinity of the epipolar line. Because we assume that there is no outside information to indicate where these preliminary matches lie along the line, we again use the hierarchical technique to search out and refine the match. If relative camera parameters have been derived, LMATCH is used on the second most interesting point in each grid cell and on any already-matched points that C2MODEL indicated were unreliable; the results of this mode are shown in Figure 5. If the true camera parameters have been supplied, LMATCH is used on the two most interesting points in each grid cell; these results are shown in Figure 6.

2.3 Anchored Matching

Once several reliable matches have been found, they can be used as "anchor" points for further matching. Our basic technique for this again uses the grid cells in the image. A given point will lie in some grid cell; the closest matched point(s) will lie in that cell or in one of the 8 neighboring cells. Under the assumption that the world is generally continuous, a point would be expected to have a disparity similar to that of its neighbors. Thus, the disparity at a point is expected to lie in the interval of the disparities of the well-matched points in the current and neighboring cells. This disparity interval is used along with the epipolar constraint to perform a very local search for the match to a point. Note that a point is considered to be well-matched if it has a correlation above a user-settable absolute threshold, usually .5, and above a variable threshold, based on the autocorrelation function around the point in the first image (see Table 1 for examples); in addition, a well-matched point cannot deviate more than a specified distance from the epipolar line.

2.3.1 Matching the Rest of the Interesting Points

At this point in our processing, we have matched the two most interesting points in each grid cell. This is still rather sparse information, so we next invoke the module PMATCH to match the balance of the interesting points. It uses the anchored match technique described above, searching along a portion of the epipolar line, to find these matches. Figure 7 shows the results of this module. Only points found to be well-matched are recorded.

2.3.2 Matching a Grid of Points

STEREOSYS permits the user to produce matched points on a closely spaced grid, if desired. The module GMATCH also uses the anchored match technique, searching along the epipolar line, to calculate matches on a user-specified grid. Figure 8 shows the results of this module on a 20 × 20 grid. The smaller marks indicate matches in which STEREOSYS has little confidence; these are currently not recorded in the data structure, leaving holes in the grid. This points up a problem with grid matching—not all areas of an image have information suitable for matching, and forcing a match at such areas can lead to poor results.

2.4 Post Processing

Although not strictly stereo processing, there are follow-up processes which are necessary to turn stereo disparities into more meaningful 3-dimensional quantities. These processes include interpolation and terrain modelling.

2.4.1 Interpolation

Often, it is not feasible to apply correlation matching at points on a pre-determined grid. Even when grid matching is feasible, there will be areas of the images that cannot be matched, due to noise in the data, insufficient information, or changes such as moving vehicles; this will result in "holes" in the grid of terrain

data, which must be filled in somehow. And, frequently, a terrain model is desired that has its points more closely spaced than that provided by the stereo matching process. In all of these cases, interpolation of the matched data points is necessary to provide information at other points. STEREOSYS incorporates an efficient interpolation scheme [Smith, 1984], permitting the user to construct elevation data grids from either randomly spaced points or a widely spaced grid of points.

2.4.2 Terrain Modeling

Given the dense grid of matched points and the camera calibration, it is possible to derive a digital terrain model. If absolute external camera information and internal camera calibration is available, the module STERDTM can be used to create a reasonably accurate DTM, which can then be displayed with another program, DTMICP. (An example of DTMICP output is shown in Figure 9; it can also produce range images of the terrain or pictures of the original imagery "painted" on the terrain.) If the only camera information is C2MODEL's relative model, then the module RELDEPTH can be used to create a relative DTM. However, due to the many over-simplifications and the computational instability of the relative camera model, such relative DTMs are of very low accuracy, and their use is discouraged.

3 Evaluation

Evaluation of the accuracy of STEREOSYS is difficult, as there do not seem to exist stereo data sets with known ground truth against which to compare our results. We do, however, have the results of an interactive stereo compilation algorithm called Digital Interactive Mapping Program (DIMP), produced and operated by the U.S. Army Engineer Topographic Laboratories (ETL) [Norvelle, 1981]. It should be noted, however, that ETL's results were obtained by an interactively coached process, which was run on a 5 × 5 grid in the 2048 × 2048 images of the Phoenix data set, and which used correlation windows warped to account for the local steepness of the terrain, while ours were obtained by a fully automatic process that ran on randomly spaced interesting points in the 1024 × 1024 images without warping. Comparing them is a little like comparing apples and oranges, but we did so in the following manner.

Comparisons were made only for those points for which STEREOSYS recorded an answer. Points were said to have the same answer if the STEREOSYS result and the result at the closest DIMP grid point (scaled into the 1024 × 1024 image in which STEREOSYS produced its results) were within one pixel of having the same disparity. Points about which there was disagreement were examined manually. An analyst looked at both results, overlaid on the images at a variety of resolutions, both monocularly and using a stereoscopic viewer, then decided which algorithm appeared to be in error and, based on experience with correlation algorithms, attempted to determine why the mistake had been made.

On the Phoenix data set, STEREOSYS found 5545 "interesting points," of which it thought it could reliably match 4676. Of these, only 43 disagreed significantly with the DIMP results for nearby points. Closer examination showed 15 of these to be uncorrected DIMP errors, 15 were STEREOSYS errors, 5 were points on which both systems appear to have made errors, and 8 were points for which the analyst could not determine which system was in error. In most of the cases, the DIMP errors seemed to result from its algorithm having drifted gradually off track (usually starting in an area with little information), and its operator not catching it soon enough. The STEREOSYS approach of first providing a context in which to work, so that the code interpolates disparities, instead of extrapolating them, should remedy this problem. Most of the STEREOSYS errors (and almost all of the points for which the analyst could not determine which algorithm was at fault) appeared to have resulted from an inappropriate threshold on the interest value: STEREOSYS was trying to match areas in which there was not enough information to make reliable matches. Some of the STEREOSYS errors were due to not using warped correlation windows to account for the slopes. In these cases, most of the information in a window would be in a corner of the window, so the disparity that was calculated was that of the corner, not the center of the window; using warped correlation or exponentially weighted interest operators and correlation windows [Quam, 1984] would solve this problem. A fair number of the mistakes (particularly the ones in which both systems arrived at different wrong answers) were because of artifacts in the data—film grain, scratches, lint, hairs, fiducial marks, and the like; we are a long way from being able to understand, let alone automate, the human ability to identify offending objects and then ignore them in processing stereo data.

STEREOSYS has also been used on several other data sets in our "challenge data base", described in Hannah [1985-b]. For data sets with no DIMP results, a much smaller number of points were matched. These were then compared with the human viewer's perception of what were the correct matches. Only the more blatant mistakes were detected and further analyzed; the results of which are presented in Hannah [1985-a].

4 Discussion

Our objective in constructing STEREOSYS was to implement a state-of-the-art, area-based system for stereo compilation operating on aerial photography. Along the way, we hoped to remedy some of the obvious problems we had seen with existing systems, such as DIMP's tendency to extrapolate itself off track. In this we have succeeded.

Because STEREOSYS uses fairly independent judgment on each match, it tends to avoid the problems we have seen in the DIMP results; indeed, on the Phoenix data set, STEREOSYS was able to duplicate DIMP's correct results (for the points tried) and rectify a number of DIMP's mistakes. Although it happens rarely, it is still possible for STEREOSYS to make mistakes in the early stages of its processing, then propagate these mistakes into later matches. To avoid this, more work needs to be done on algorithms for detecting improperly matched points, so they can be removed before further processing.

The major criticism we have heard of STEREOSYS is that it produces matches at randomly spaced points (only where adequate information is present), when what is usually wanted is a closely spaced regular grid of elevation points, regardless of image content. So far, attempts at blindly interpolating the disparity data (ignoring the image data) as reported in Smith [1984] have proven less than satisfying. Marriage of the STEREOSYS techniques with something like DIMP, or with hierarchical warp correlation [Quam, 1984], or with image intensity-based interpolation [Smith, 1985 or Baker, 1982] might be profitable.

We have performed one experiment as a preliminary study in how to integrate the strengths of STEREOSYS with those of an edge-based matcher. The results of STEREOSYS were

used as seeds for an edge-based matching system [Baker, 1982], which propagated these matches along the nearby zero-crossing contours, then did one iteration of edge matching. Because determining disparity constraints is a large part of the edge-based matcher's processing, introducing this information from STEREOSYS's results produced a significant reduction in computation time used by the edge-based matcher. The number of matched points also increased by about an order of magnitude over the results of STEREOSYS alone. Although we have not yet finished a quantitative evaluation of these match accuracies, a qualitative analysis indicates that the results from the combined technique are significantly more accurate than the results of the edge-based system alone.

Overall, we have found that STEREOSYS performs credibly on the low-resolution aerial imagery for which it was designed. It has difficulties when processing areas that violate its premises about the continuity of the world, but linking it with an edge-based matcher (which would excel in these types of areas) seems to be a promising approach.

Acknowledgements

The research reported herein was supported by the Defense Advanced Research Projects Agency under Contract MDA903-83-C-0027, which is monitored by the U.S. Army Engineer Topographic Laboratory. The views and conclusions contained in this paper are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or of the United States Government.

I would like to thank Harlyn Baker, Robert Bolles, Martin Fischler, Lynn Quam, and Grahame Smith for their support on this project.

References

Baker, H. Harlyn, 1982. "Depth from Edge and Intensity Based Stereo," Ph.D. Thesis, Stanford University Computer Science Department Report STAN-CS-82-930, September 1982.

Burt, Peter J., 1981. "Fast Filter Transforms for Image Processing," *Computer Graphics and Image Processing*, Vol. 16, pp. 20-51, 1981.

Gennery, Donald B., 1980. "Modelling the Environment of an Exploring Vehicle by means of Stereo Vision," Ph.D. Thesis, Stanford University Computer Science Department Report STAN-CS-80-805, June, 1980.

Hannah, Marsha Jo, 1974. "Computer Matching of Areas in Stereo Images," Ph.D. Thesis, Stanford University Computer Science Department Report STAN-CS-74-438, July, 1974.

Hannah, Marsha Jo, 1980. "Bootstrap Stereo," *Proceedings: Image Understanding Workshop*, College Park, MD, April, 1980, pp. 201-208.

Hannah, Marsha Jo, 1985-a. "Evaluation of STEREOSYS vs. Other Stereo Systems", SRI International Artificial Intelligence Center Technical Note 365, October, 1985.

Hannah, Marsha Jo, 1985. "The Stereo Challenge Data Base", SRI International Artificial Intelligence Center Technical Note 366, October, 1985.

Moravec, Hans P., 1980. "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover," Ph.D. Thesis, Stanford University Computer Science Department Report STAN-CS-80-813, September, 1980.

Nishihara, H. Keith, and Tomaso Poggio, 1983. "Stereo Vision for Robotics," *Proceedings of the International Symposium of Robotics Research*, Bretton Woods, NH, September, 1983.

Norvelle, F. Raye, 1981. "Interactive Digital Correlation Techniques for Automatic Compilation of Elevation Data," U.S. Army Engineer Topographic Laboratories Report ETL-0272, October, 1981.

Panton, Dale J., 1978. "A Flexible Approach to Digital Stereo Mapping," *Photogrammetric Engineering and Remote Sensing*, Vol. 44, No. 12, pp. 1499-1512.

Quam, Lynn H., 1971. "Computer Comparison of Pictures," Ph.D. Thesis, Stanford University Computer Science Department Report STAN-CS-71-219, May, 1971.

Quam, Lynn H., 1984. "Hierarchical Warp Stereo," *Proceedings: Image Understanding Workshop*, New Orleans, LA, October, 1984, pp. 149-156.

Smith, Grahame B., 1984. "A Fast Surface Interpolation Technique," *Proceedings: Image Understanding Workshop*, New Orleans, LA, October, 1984, pp. 211-215.

Smith, Grahame B., 1985. "Stereo Reconstruction of Scene Depth," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, June 9-13, 1985.

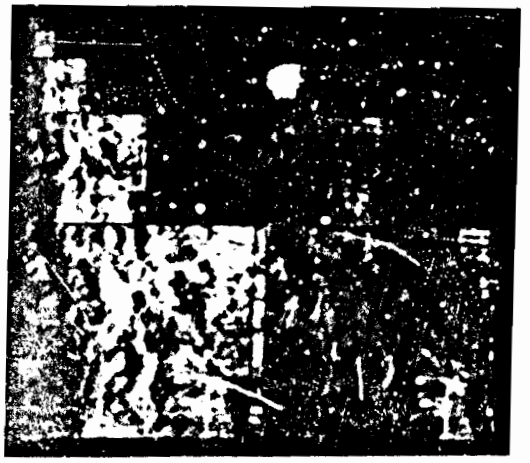


Figure 1-Reduction Image Hierarchy.

Figure 2-Interesting Points, Ranked by Grid Cell.

Figure 3-Hierarchical Match of an Interesting Point.

Table 1-Hierarchical Correlations for Point in Figure 3.

Image size	Point 1	Point 2	Correlation	Autocorrelation
16x16	(9, 11)	(9, 11)	0.140452	0.677717
32x32	(19, 23)	(19, 23)	0.384883	0.437053
64x64	(38, 46)	(37, 46)	0.738581	0.738427
128x128	(77, 92)	(76, 92)	0.929933	0.885289
256x256	(154, 184)	(153, 184)	0.954606	0.918228
512x512	(308, 369)	(306, 369)	0.916062	0.929428
1024x1024	(616, 738)	(612, 737)	0.750448	0.932947
2048x2048	(1232, 1476)	(1222, 1475)	0.341622	0.790917

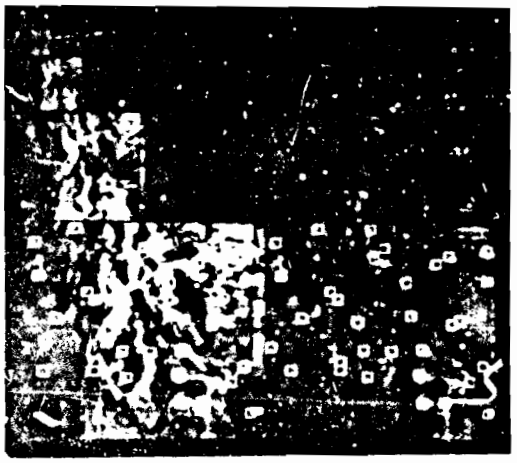


Figure 4-Results of Unstructured Hierarchical Matching of Most Interesting Point in Each Grid Cell.

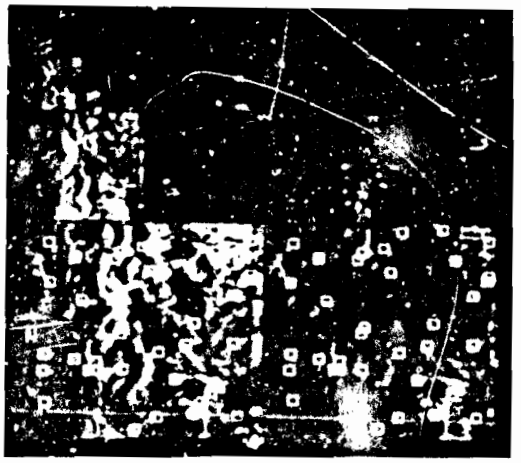


Figure 5-Results of Epipolar Hierarchical Matching of Second Most Interesting Point in Each Grid Cell.

Figure 6-Results of Epipolar Hierarchical Matching of Two Most Interesting Points in Each Grid Cell.

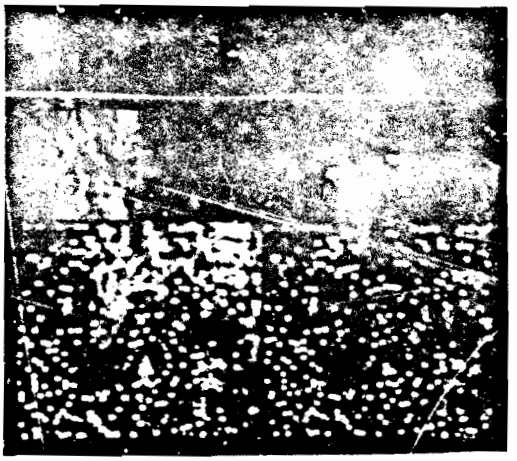


Figure 7-Results of Anchored Epipolar Matching of Remaining Interesting Points.

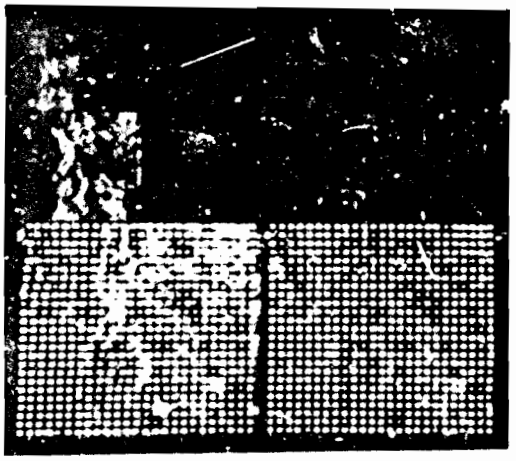


Figure 8-Results of Anchored Epipolar Matching of a Grid of Points.

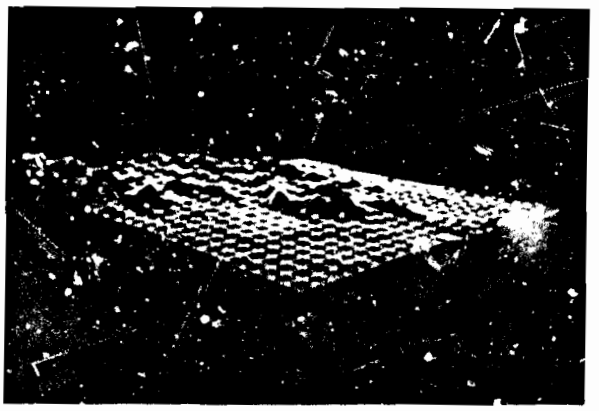


Figure 9-A View of the Resulting Digital Terrain Model.

GC970145-21

CONCURRENT MULTILEVEL RELAXATION

Demetri Terzopoulos

MIT Artificial Intelligence Lab.
545 Technology Square
Cambridge, MA 02139

Abstract

Multigrid relaxation techniques lead to highly efficient iterative algorithms that are well suited to the optimization solution of problems arising in low level image understanding. Standard schemes for coordinating multilevel relaxation processes are designed for sequential machines and leave all but a single level idle at any time. This paper develops a concurrent multigrid coordination strategy which maintains simultaneous activity in all levels. Consequently, concurrent multigrid relaxation can fully exploit the processing power offered by the new generation of massively parallel, fine-grained hardware with local interprocessor connections.

1. Introduction

It has become increasingly evident in recent years that explicit recognition of the presence of multiple scales of resolution leads to approximation methods and associated algorithms of great power and efficiency. As a consequence, machine vision researchers are intensively developing various approaches to multiresolution image processing and analysis [Rosenfeld, 1984]. Among these is an approach that adapts ideas central to a class of iterative numerical techniques known as multigrid relaxation methods [Terzopoulos, 1983]. Efficient algorithms based on these methods have been developed for a number of computational problems that arise in the early stages of image understanding [Terzopoulos, 1983, 1986a; Glazer, 1984].

Multigrid relaxation techniques were originally targeted to the efficient numerical solution of elliptic partial differential equations [Hackbusch and Trottenberg, 1982]. Equations of this kind express necessary conditions for the stationary points of variational principles. Stated as the optimization of certain objective functionals (i.e., energy functionals $\mathcal{E}(v)$), the latter, it turns out, arise naturally in early vision from the regularization of a wide range of mathematically ill-posed inverse visual problems (refer, e.g., to the MIT progress report by Poggio *et al.* in this proceedings and to [Terzopoulos, 1986b]). Because of this relationship, the utility of multigrid methodology in early vision is potentially very broad.

Another attraction of multigrid methods is that they can be designed to require only local, parallel computations. Hence, they are implementable on the locally interconnected, parallel computer architectures now made possible by VLSI technology; in particular, the massively parallel processors that have been conceived for image processing and AI applications [Batcher, 1980; Hillis, 1985].

In this paper, we develop a concurrent multigrid relaxation scheme for early visual processing which, unlike standard multigrid algorithms, is designed to fully utilize the processing power offered by this new generation of distributed computers.

1.1. Conventional Multigrid Processes

From a Fourier analysis point of view, since relaxation is a local operation, it can compute efficiently only a confined range of spatial components of the desired discrete solution. The useful efficiency range is determined by the resolution of the grid. More precisely, the shortest wavelength components of the approximation error function (those on the order of the internode spacing) are annihilated quickly, whereas the longer wavelength components persist over many iterations [Brandt, 1977].

Multigrid methods significantly extend the efficiency range of relaxation through the use of exponentially tapered multiresolution grid hierarchies. In practice, one uses a set of regular, discrete grids with successive doubling of internode separations from one level to the next coarser level (i.e., successive halving of resolution). A portion of the standard grid hierarchy is illustrated in Fig. 1. In principle, it can be mapped onto regularly interconnected VLSI architectures, and in a fully parallel multigrid implementation, each node would represent an individual processing element.

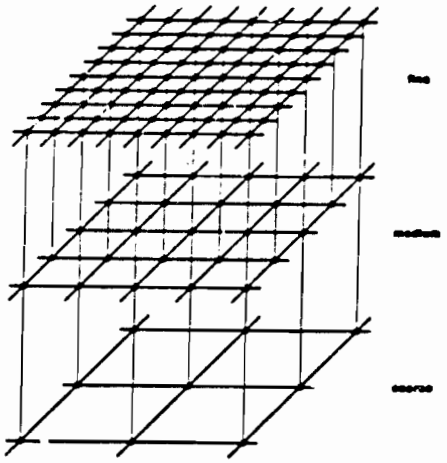


Figure 1. Typical multigrid organization. A portion of three levels of the 2:1 multigrid hierarchy is illustrated. Only nearest-neighbor interprocessor connections are shown.

Relaxation in either Jacobi (parallel) or Gauss-Seidel (serial) form is the basic iterative solution method employed on each level. In conjunction with intralevel relaxation processes, a series of coarse-to-fine (extension) and fine-to-coarse (restriction) processes permit transfer of information between adjacent levels. These subprocesses are coordinated to compute the multigrid solution.

1.2. Recursive vs Concurrent Multigrid Coordination

Conventional multigrid schemes employ recursive coordination strategies [Brandt, 1977]. The solution of a discrete problem on the finest grid requires the solution of a sequence of related problems on coarser levels which, in turn, require the solution of related problems on still coarser levels. The recursion is carried to a depth where the grids are sufficiently coarse that the expense in computing solutions becomes trivial. Further, the recursive multigrid scheme can be applied starting from the coarsest level and proceeding successively to the finest level, using the results at each level as an initial approximation for the next level (this is known as nested relaxation; see, e.g., [Terzopoulos, 1986a] for the algorithm).

The recursive multigrid coordination strategies were targeted at serial computers. Although effective on uniprocessors, they are not designed to make optimal use of all available processing elements in a massively parallel machine, even when a spatially parallel Jacobi relaxation is employed. This is because most of the time is spent performing relaxations on only a single level, while processors on the other levels remain idle, with the rest of the time spent transferring results between pairs of adjacent levels. Very poor utilization of processors results on average, since a significant percentage of the iterations and transfers are performed on the coarser levels.

This deficiency has prompted proposals for increasing the level of parallelism. One approach is via clever multilevel decompositions of the problem, such that a concurrent coordination of multigrid processes does not become self defeating. Brandt [1981] pursued some early ideas along these lines in the context of vector supercomputers, which offer a rather low level of parallelism, but the convergence rates that he reported were not impressive. Gannon and Rosendale [1982] proposed multilevel decompositions that render the multigrid solution better suited to massively parallel computers. In the context of visual surface reconstruction, Terzopoulos [1984] suggested a multilevel decomposition similar to Gannon's, but the idea remains to be implemented. Kuo [1985] extended Gannon's approach to make more systematic use of digital filtering theory.

While fully concurrent multigrid coordination across all levels is clearly desirable in general, the need to solve piecewise continuous reconstruction problems in early vision [Terzopoulos, 1986b] raises particular computational issues which must be resolved when designing multigrid algorithms for vision. In particular, there arises a need to process irregularly structured, sparse data input at multiple resolutions. The data generally includes both constraints and discontinuities.

The concurrent multigrid algorithms proposed in the literature are not designed to handle visual data. As argued by Kuo, proper application of the decomposition approach requires the use of relatively large support filters. Unfortunately, such filters require continuous regions and geometrically regular boundary conditions. This virtually prohibits their application to visual problems involving irregularly occurring constraints and discontinuities.

The present paper takes a variational approach to concurrent multigrid relaxation which promises, in principle, to resolve these difficulties. The formulation is based on a multilevel objective functional which is developed in Section 2. This leads, in Section 3, to a relaxation subprocess for each level, which includes not only neighboring nodes on that level, but couples nearby nodes on the adjacent coarser and finer levels as well. The subprocesses are driven concurrently across the multiple levels. Section 4 proposes a dynamic strategy for adjusting the coupling strengths between relaxation processes during the iterative process in order to achieve a common multilevel objective.

2. Concurrent Multilevel Objective

The objective of multilevel relaxation is to efficiently compute discrete, multiresolution approximations to the solutions of given continuous problems. Formally speaking, discrete approximations characterize optima of a multilevel objective functional. The approximations are constructed in nested collections of finite dimensional approximating spaces.

2.1. Nested Finite Element Spaces

Suppose, in particular, that the finite dimensional approximating spaces are a family of L finite element spaces $\{S^l\}_{l=1}^{L}$, nested such that $S^l \subset S^{l+1}$. It is convenient for the size of the elements of S^l to be given by $h^l = 2^{L-l}h^L$, relative to the fundamental size h^L of elements in the finest space S^L. We will refer to S^{l-1} and S^{l+1} as being adjacent to S^l (note that S^1 and S^L have only one adjacent space).

The basis functions of finite element spaces have local support; that is, they involve only spatially proximal nodal variables. When the nodal variables form a grid, it is convenient to write the basis functions of S^l as $\phi^l_{i,j}$, where the subscripts i,j identify a node of the grid. One can then express a generic function $v^l \in S^l$ as

$$v^l = \sum_{i=1}^{M}\sum_{j=1}^{N} v^l_{i,j}\phi^l_{i,j},$$

where $v^l_{i,j} \in \Re$ denotes a nodal variables associated node i,j. The family of such functions is denoted $\mathbf{v} = (v^1, \ldots, v^L) \in S^1 \times \cdots \times S^L$.

2.2. The Multilevel Objective Functional

Given the above definitions, the multilevel objective functional $\mathbf{E}(\mathbf{v}) : S^1 \times \cdots \times S^L \rightarrow \Re^L$ may be defined as

$$\mathbf{E}(\mathbf{v}) = \begin{bmatrix} \mathcal{E}^1(v^1) \\ \vdots \\ \mathcal{E}^l(v^l) \\ \vdots \\ \mathcal{E}^L(v^L) \end{bmatrix} + \frac{1}{2}\mathbf{C}(\mathbf{v}).$$

The first term is derived from the energy functional $\mathcal{E}(v)$ for the problem at hand, whose discrete approximation in the finite element space S^l is denoted by $\mathcal{E}^l(v^l) : S^l \rightarrow \Re$. See [Terzopoulos, 1983, 1985, 1986a] for detailed derivations of discrete functionals on finite element spaces for a number of vision problems. The vector functional $\mathbf{C}(\mathbf{v})$ serves to couple the approximation computed in each space S^l to its adjacent spaces.

2.3. The Multilevel Coupling Functional

The multilevel coupling functional can be written as

$$\mathbf{C}(\mathbf{v}) = \begin{bmatrix} \mu^1 \|v^1 - \Pi^1[v^2]\|^2 \\ \vdots \\ \kappa^l \|v^l - \Gamma^l[v^{l-1}]\|^2 + \mu^l \|v^l - \Pi^l[v^{l+1}]\|^2 \\ \vdots \\ \kappa^L \|v^L - \Gamma^L[v^{L-1}]\|^2 \end{bmatrix},$$

where it is convenient for $\|\cdot\|$ to be the Euclidean norm. The constituent functionals involving κ factors impose a coarse-to-fine coupling between adjacent spaces, while those involving μ factors impose the converse fine-to-coarse coupling. All coupling factors are real, non-negative numbers, and their magnitudes determine the strength of the interlevel couplings. The coupling functional also involves two sets of mappings (denoted Γ and Π) to each S^l from its adjacent finite element spaces.

2.4. Interlevel Mappings

The interlevel mappings perform the required interlevel changes of bases. The coarse-to-fine basis change is accomplished by an injection mapping

$$\Gamma^l : S^{l-1} \rightarrow S^l; \qquad l = 2, \ldots, L,$$

while the converse fine-to-coarse change is accomplished by a projection mapping

$$\Pi^l : S^{l+1} \rightarrow S^l; \qquad l = 1, \ldots, L-1.$$

For grids, the interlevel mappings take the general

form

$$I^l(v^{l-1})_{i,j} = \sum_{m,n=1}^{M,N} \iota^{l-1}_{i,j,m,n} v^{l-1}_{i,j},$$

$$\Pi^l(v^{l-1})_{i,j} = \sum_{m,n=1}^{M,N} \pi^{l-1}_{i,j,m,n} v^{l-1}_{i,j},$$

which corresponds to matrix multiplications on the nodal variables. In practice, local support mappings are employed, which means that the coefficients $\iota^{l-1}_{i,j,m,n}$ and $\pi^{l-1}_{i,j,m,n}$ are nonzero only within some neighborhood of node i,j. This corresponds to multiplications with sparse matrices. Indeed, it is most natural (though not absolutely necessary) to specify the coefficients according to the local interpolant of the finite element space in which the argument function resides.

3. Concurrent Multilevel Relaxation

As a natural extension to numerically solving single-level variational principles (Terzopoulos, 1983), the necessary condition for optimizing the multilevel objective functional — i.e., characterizing $\mathbf{u} = \inf_{\mathbf{v}} \mathbf{E}(\mathbf{v})$ according to the vanishing of the gradient $\nabla \mathbf{E}(\mathbf{v})$ — results in a multilevel system of algebraic equations. Provided that the given energy functional $\mathcal{E}(v)$ derives from a well-posed variational principle, the necessary condition will also be sufficient, because the added multilevel coupling functional is quadratic and positive definite due to the presence of norms. The algebraic system will then have a unique solution, which the multilevel relaxation process approximates iteratively.

Multilevel relaxation may be viewed as a vector intralevel mapping

$$\mathbf{R} : S^1 \times \cdots \times S^L \to S^1 \times \cdots \times S^L.$$

Its form at a generic node i,j is determined by setting to zero the partial derivative of the multilevel objective functional with respect to $v^l_{i,j}$, for $l = 1, \ldots, L$. The following general form results:

$$\mathbf{R}(\mathbf{v})_{i,j} = \begin{bmatrix} r^1(v^1)_{i,j} + \mu^1(v^1_{i,j} - \Pi^1 v^2_{i,j}) \\ \vdots \\ r^l(v^l)_{i,j} + \kappa^l(v^l_{i,j} - I^{l-1} v^{l-1}_{i,j}) + \mu^l(v^l_{i,j} - \Pi^l v^{l+1}_{i,j}) \\ \vdots \\ r^L(v^L)_{i,j} + \kappa^L(v^L_{i,j} - I^{L-1} v^{L-1}_{i,j}) \end{bmatrix}$$

The $r^l(v^l)_{i,j}$ terms correspond to the relaxation operator obtained for a standard single-level minimization of $\mathcal{E}(v^l)$, and take the general form

$$r^l(v^l)_{i,j} = \sum_{m,n=1}^{M,N} \rho^l_{i,j,m,n} v^l_{i,j},$$

where the coefficients $\rho^l_{i,j,m,n}$ are nonzero only for nodes m,n which are proximal to node i,j.

The information flow within the hierarchy implied by the concurrent multilevel relaxation scheme is illustrated in Fig. 2

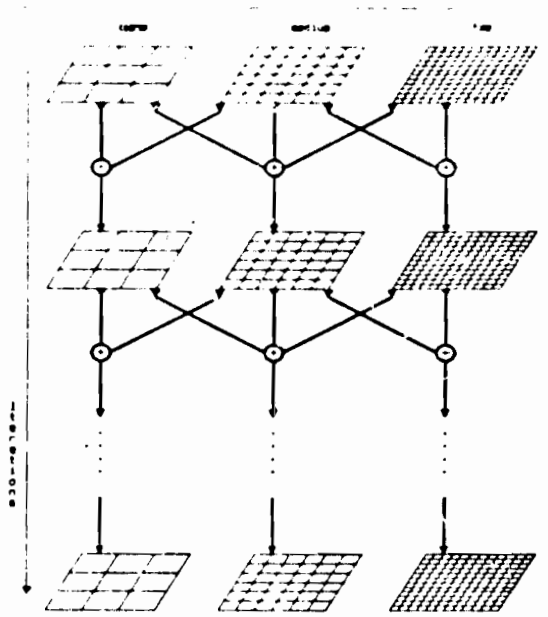


Figure 2. Information flow within the multigrid hierarchy.

4. Dynamic Coupling

There remains the issue of making appropriate choices for the coupling strengths κ^l and μ^l so as to obtain useful multilevel solutions. As a general rule, the relaxation processes on the coarser scales suffer from increasingly large discretization errors, but converge to the coarse solution relatively quickly. Conversely, those on the finer scales are increasingly accurate, but exhibit a substantially slower response.

On the one hand, by setting the κ^l to some intermediate value, the relaxation processes are coupled to

coarser ones by way of intermediate processes, so that the fast response characteristics of the coarser relaxation processes is, to some degree, extended to the entire multigrid hierarchy. The coarse-to-fine coupling can be increased to heighten the effect, but beyond a certain point the poor accuracy of the coarser levels corrupts the solution computed in the finest levels.

On the other hand, by setting the μ^l to some intermediate value, coarser relaxation processes are coupled to the finer ones so that the higher accuracy of the finer approximations permeates the whole multigrid hierarchy. However, as the fine-to-coarse coupling is tightened, the multigrid hierarchy tends to be infected by the slow response characteristics of the finer relaxation processes.

Dynamic coupling resolves the dilemma. By appropriately adjusting the coupling strengths during the iterative process, the multigrid hierarchy simultaneously inherits a fast response and a consistently high accuracy. A general strategy for adjusting the coupling factors is the following. Initially there is a strong coarse-to-fine interaction. This accelerates the convergence rate of the finer relaxation processes during the early iterations, when the approximation is rather poor. This interaction decays as the approximation improves, and is replaced by a strengthening fine-to-coarse interaction, which eventually enables the accurate approximations computed on the finer levels to dominate and improve the accuracy of the coarser approximations. In particular, one can choose

$$\kappa^l(t) = \kappa_0^l a^t; \qquad \mu^l(t) = \mu_0^l(1 - a^t),$$

where t is the iteration index and a is a decay in the range $0 < a < 1$, to obtain curves such as those illustrated in Fig. 3.

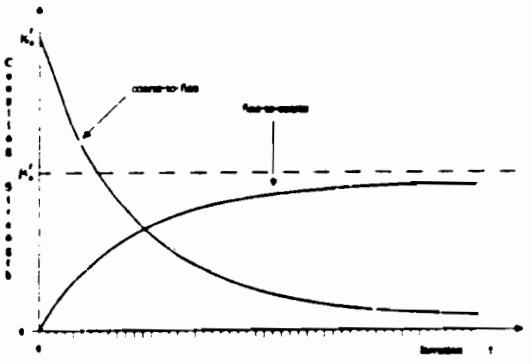


Figure 3. Dynamic coupling. The magnitude of the coupling factors varies during the iterative process.

5. An Example

For a concrete example of a concurrent multilevel relaxation process, consider the regularization solution of the surface reconstruction problem using a thin plate surface spline (see Terzopoulos, 1983). A Jacobi version of the relaxation formula that was obtained for an interior node of the level l grid is given by

$$r^l(v^l)_{i,j} = \Big[8\,(v^l_{i-1,j} + v^l_{i+1,j} + v^l_{i,j-1} + v^l_{i,j+1}) - 2\,(v^l_{i-1,j-1} + v^l_{i+1,j-1} + v^l_{i-1,j+1} + v^l_{i+1,j+1}) - 1\,(v^l_{i-2,j} + v^l_{i+2,j} + v^l_{i,j-2} + v^l_{i,j+2}) + (h^l)^2 \alpha^l d^l_{i,j} \Big] \Big/ \left(20 + (h^l)^2 \alpha^l\right),$$

where α^l is the constraint parameter associated with the scattered depth constraints $d^l_{i,j}$ input on level l.

The above formula may be used in the concurrent multilevel relaxation process

$$v_{i,j}^{(t+1)} = \mathbf{R}(v)^{(t)}_{i,j},$$

where t is the iteration index, in conjunction with a simple coarse-to-fine coupling term

$$\kappa^l(v^l_{i,j} - v^{l-1}_{\frac{i}{2},\frac{j}{2}}); \qquad \text{for } i,j \text{ odd, and zero otherwise,}$$

and the analogous fine-to-coarse coupling term

$$\mu^l(v^l_{i,j} - v^{l+1}_{2i-1,2j-1}); \qquad \text{for all } i,j.$$

The effect of these strictly local expressions can be interpreted physically as unilateral springs with stiffnesses κ^l and μ^l, which couple at coincident grid coordinates the thin plate splines on adjacent levels (imagine the vertical connections in Fig. 1 to be the springs). The natural smoothing of the relaxation process permits the even-numbered nodes to remain uncoupled from the adjacent coarser level.

The reader should bear in mind, that analogous relaxation formulas result from more sophisticated surface models which allow arbitrary discontinuity boundaries to be introduced on the grids [Terzopoulos, 1985; 1986b], as well as from related image analysis problems [Terzopoulos, 1986a].

6. Summary and Discussion

We have developed a new multilevel relaxation strategy that exploits a greater degree of parallelism. In con-

trast to conventional, recursive coordination schemes, the new coordination strategy is fully concurrent: it maintains processors on all levels busy performing simultaneous relaxation operations.

The concurrent coordination strategy aims to optimize a multilevel objective functional, each of whose terms has three components: (1) a discrete version of the given functional on each level of a multigrid hierarchy, (2) an additive functional coupling each level (except the finest) to the next finer level, and (3) an additive functional coupling each level (except the coarsest) to the next coarser level.

The interlevel coupling functionals are designed so that the scheme will be convergent. They involve coupling factors that are modified during the iterative process such that there is an initially strong but gradually weakening coarse-to-fine interaction, which accelerates convergence, and an initially weak but gradually strengthening fine-to-coarse interaction, which ultimately yields consistent accuracy on all levels.

We have implemented a concurrent multigrid algorithm for the problem of computing visible surface representations as formulated in [Terzopoulos, 1985]. Preliminary experiments with the algorithm are encouraging. The efficiency of the concurrent algorithm on a uniprocessor is comparable to that of its conventional multigrid counterpart (the efficiency of the latter is studied in [Terzopoulos, 1984]). We also noted that the concurrent algorithm is significantly easier to implement than its conventional counterpart.

Among the many issues that beckon further study are:

- The implementation of concurrent multigrid schemes in parallel hardware. Some analysis of the implications can be found in [Brandt, 1981] and [Gannon and Rosendale, 1982].
- The investigation of coupling functional that are not based on common norms. This will lead to more sophisticated methods for fusing information across scales.
- The use of more complicated interlevel mappings in the coupling functional. This will suggest, for example, nontrivial ways of treating discontinuities across scales.
- Reformulating the concurrent multigrid approach to generate distributed multiresolution representations, such as the relative depth representations suggested in [Terzopoulos, 1984]. One possibility, which draws from the previous point, is to use interlevel mappings that constitute weighted sums of nodal variables on a series of coarser levels (along with residual input data differences as constraints).

Acknowledgements

I thank Michael Brady for challenging me to develop a concurrent multilevel relaxation algorithm for vision.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's Artificial Intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-80-C-0505 and the System Development Foundation.

References

Batcher, K.E., [1980], "Design of a massively parallel processor," *IEEE Trans. Computers*, **C-26**.

Brandt, A., [1977], "Multi-level adaptive solutions to boundary-value problems," *Math. Comp.*, **31**, 333–390.

Brandt, A., [1981], "Multigrid solvers on parallel computers," *Elliptic Problem Solver*, M.H. Schultz (ed.), Academic Press, New York, 39–83.

Gannon, D., and Rosendale, J.V., [1982], Highly parallel multigrid solvers for elliptic PDEs: An experimental analysis, ICASE, NASA Langley Research Center, Hampton, VA, ICASE Report 82-36.

Glazer, F., [1984], "Multilevel relaxation in low level computer vision," *Multiresolution Image Processing and Analysis*, A. Rosenfeld (ed.), Springer-Verlag, New York, 312–330.

Hackbusch, W., and Trottenberg, U., (ed.), [1982], *Multigrid Methods*, Lecture Notes in Mathematics, Vol. 960, Springer-Verlag, New York.

Hillis, W.D., [1985], The connection machine, Ph.D. thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA.

Kuo, C.C., [1985], Parallel algorithms and architectures for solving elliptic partial differential equations, MIT Lab. for Information and Decision Systems, Cambridge, MA, LIDS TH-1432.

Rosenfeld, A. (ed.), [1984], *Multiresolution Image Processing and Analysis*, Springer-Verlag, New York.

Terzopoulos, D., [1983], "Multilevel computational processes for visual surface reconstruction," *Computer Vision, Graphics, and Image Processing*, **24**, 52-96.

Terzopoulos, D., [1984], Multiresolution computation of visible-surface representations, Ph.D. thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA.

Terzopoulos, D., [1985], Computing visible-surface representations, MIT AI Lab., Cambridge, MA, AI Memo No. 800.

Terzopoulos, D., [1986a], "Image analysis using multigrid relaxation methods," *IEEE Trans. Patt. Analysis and Machine Intelligence*, **PAMI-8**, to appear.

Terzopoulos, D., [1986b], "Regularization of inverse visual problems involving discontinuities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-8, to appear.

GC970145-22

Trinocular Vision Using Photometric and Edge Orientation Constraints[1]

Victor J. Milenkovic and Takeo Kanade

Department of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

Trinocular vision is stereo using three non collinear views. It has been shown in the literature that a third view aids in the selection of matching pairs of edge points from the first two views by providing a constraint on the positions of the points. In addition to this positional constraint, this paper proposes two new constraint principles for use in determining the set of correct matches. The first principle constrains the orientations of the matched edge pixels and the second principle constrains the image intensity values in the regions surrounding the edge pixels. Statistical confidence measures and rejection thresholds are derived from these constraint principles in order to maximize the number of correct matches in the presence of error. An trinocular stereo algorithm based on these principles is described and applied to synthetic and real images with good results.

1. Introduction

A typical method of binocular stereo vision has three stages. First, the vision system extracts edges from gray level images taken from two known viewpoints. Next, the system attempts to match edge pixels in one image with edge pixels in the other image in order to determine the depth of each edge point. Finally, some form of surface fitting extends the sparse depth data to the entire image. One of the most difficult tasks in this method is to disambiguate multiple matches in the second stage. A number of papers have shown that adding a third view of the same scene can greatly serve to eliminate incorrect matches. The extra view provides a geometric constraint on the position of edge pixels [4] and [3] or on the positions of other features [1]. For correlation based matching, it has been shown that a measure based on multiple images has much higher and better defined peaks [7] than that of a two image correlation.

In this paper, two new constraints on the matched edge points are proposed. The first is a geometric constraint on the orientations of the edges which holds only when the number of cameras is greater than two. The second new principle proposed is a photometric constraint on the intensities of light measured in the vicinity of the edge points. To provide insensitivity to noise, these constraints are transformed into statistical confidence measures. These confidence measures are combined with the geometric position constraint used by the other trinocular methods to create a new matching algorithm. The use of statistical methods enables the matching algorithm to detect cases of occlusion and also to identify edges arising from occluding contours.

The trinocular matching algorithm is demonstrated on both synthetic and real data. It performs as well as a good binocular scheme and also matches horizontal edges. The algorithm works very well despite the fact that it uses none of the higher level assumptions that binocular schemes must use. In general, binocular methods require an assumption of depth continuity as well as other constraints on the disparity range in order to reduce the number of candidate matches. In order to exploit these assumptions, computationally expensive methods such as relaxation [3], dynamic programming [6], or multiple resolution matching must be used. The trinocular method demonstrated here does not require these assumptions; it looks only at individual edge pixels and thus is computationally efficient.

2. Constraint Principles

A trinocular matching algorithm can use at least three constraint principles to discriminate correct matches from incorrect ones. Two are geometric constraints on the edge pixel positions and orientations. The third is photometric constraint on the image intensity near the edge. The efficiency of the algorithm depends on the first position constraint, which is in fact the strongest of the three. The other two constraints provide additional support for correct matches. This section introduces the geometric and photometric principles behind these constraints, and section 3 describes the statistical techniques needed in order to exploit the edge orientation and image intensity constraints in the presence of noise.

In order to maximize the additional information it provides, the third camera should not lie collinear with the other two cameras. Optimally, the three cameras are positioned in an equilateral triangle. Two of the cameras, called *left* and *right* are set up side by side in the standard arrangement for two camera stereo. The third camera, the *up* camera, sits between the first two, above or below them to complete the triangle. For simplicity of calculations, the cameras should all be directed parallel to each other.

[1]This research was sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, and monitored by the Air Force Avionics Laboratory under Contract F33615-78-C-1551. The views and conclusions in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

2.1 Position Constraint

The trinocular position constraint is a simple extension of the one which exists for the binocular matching algorithm. A given edge pixel in the left image can match only edge pixels in the right image which lie on an epipolar line. Since the left and the right cameras differ in horizontal position only, the epipolar lines are the scan lines of the images. But the same constraint principle holds for the left and up cameras and the right and up cameras. In these cases the epipolar lines do not correspond to scan lines because they are inclined by 60 and 120 degrees, but the epipolar constraint holds nevertheless. The two lines meet in the up image at an angle of 60 degrees and therefore always determine an unambiguous location for the third edge pixel in a matched triple. Ordinarily, an edge pixel in the left image and an edge pixel in the right image cannot match unless a corresponding edge pixel exists at the correct location in the up image, and thus the third image provides a geometric position constraint not available to the binocular algorithm. The trinocular algorithm requires little or no search in the up image, and thus it is roughly as efficient as the simplest binocular algorithm.

2.2 Orientation Constraint

An edge detection algorithm can respond only to changes in intensity in the image. These intensity changes can have a number of causes all having to do with changes in the surface of the object as seen by the camera.

- The surface may have an abrupt change in reflectivity, such as at the boundary of two regions with different colors.
- The surface normal may be discontinuous as at a polyhedral edge.
- The surface may vanish because it is occluded by another surface or it occludes itself.

In general these conditions can occur in any combination, but if there is no reflectivity or surface normal discontinuity, the resulting edges depend on the viewpoint and thus are useless for matching. If either of the first two conditions hold, then there is a viewpoint independent curve in space which is the boundary of discontinuity.

The edge pixels in an image correspond to points on the contour of discontinuity. The measured orientations of the edges are the projections of the tangent lines to the contour. Suppose the edge orientation vectors $\mathbf{Q}_L$, $\mathbf{Q}_R$, and $\mathbf{Q}_U$ are projections of the contour tangent vector $\mathbf{v}$ as in figure 2-1. Let the vector displacements from camera to image plane point be $\mathbf{D}_L$, $\mathbf{D}_R$, and $\mathbf{D}_U$. The vector $\mathbf{v}$ must lie in the plane determined by $\mathbf{D}_L$ and $\mathbf{Q}_L$, the plane determined by $\mathbf{D}_R$ and $\mathbf{Q}_R$, and the plane determined by $\mathbf{D}_U$ and $\mathbf{Q}_U$. Therefore the normal vectors to these three planes, $\mathbf{N}_L$, $\mathbf{N}_R$, and $\mathbf{N}_U$, must lie in the plane perpendicular to $\mathbf{v}$, and thus the triple product of these three vectors must be zero.

$$\mathbf{N}_L = \frac{\mathbf{D}_L \times \mathbf{Q}_L}{|\mathbf{D}_L \times \mathbf{Q}_L|}, \quad \mathbf{N}_R = \frac{\mathbf{D}_R \times \mathbf{Q}_R}{|\mathbf{D}_R \times \mathbf{Q}_R|}, \quad \mathbf{N}_U = \frac{\mathbf{D}_U \times \mathbf{Q}_U}{|\mathbf{D}_U \times \mathbf{Q}_U|},$$

$$[\mathbf{N}_L\ \mathbf{N}_R\ \mathbf{N}_U] = 0 \quad \text{(triple product)}.$$

2.3 Photometric Constraint

As has been remarked above, not all edge pixels are useful for purposes of matching. In order to be useful, an edge pixel must correspond to a point on some viewpoint independent contour, such as the boundary between surfaces of differing reflectivity or surface normal. Of course, one of the surfaces may not be visible as in the case of an occluding contour, but at least one, and usually both, of the surfaces meeting at the contour are visible from all three cameras. In general, an edge pixel splits the region nearby in the image into two parts, one on each side of the edge. One side has somewhat depressed intensity, the other, somewhat elevated intensity. These *darker* and *lighter* sides correspond to the surfaces of different orientation or reflectivity on the object that meet at the contour that causes the edge pixel. Moreover, the darker side regions of matching edge pixels from different images will correspond to the same surface on the object. The lighter side regions will correspond to another common surface.

Suppose we sample the images near edge pixels which are supposed to match. As in figure 2-2, the k sample points on the darker side of the pixel in the left image have intensities $I_1^{L-}, \ldots, I_k^{L-}$, the sample points on the lighter side of left pixel have intensities $I_1^{L+}, \ldots, I_k^{L+}$, and so on. Define the means of the sampled points as follows,

$$\mu_L^- = \frac{1}{k}\sum_{i=1}^{k} I_i^{L-}, \quad \mu_L^+ = \frac{1}{k}\sum_{i=1}^{k} I_i^{L+},$$

$$\mu_R^- = \frac{1}{k}\sum_{i=1}^{k} I_i^{R-}, \quad \mu_R^+ = \frac{1}{k}\sum_{i=1}^{k} I_i^{R+},$$

$$\mu_U^- = \frac{1}{k}\sum_{i=1}^{k} I_i^{U-}, \quad \mu_U^+ = \frac{1}{k}\sum_{i=1}^{k} I_i^{U+}.$$

If the darker sides do indeed correspond to the same surface on the object, then one would expect the means μ_L^-, μ_R^-, μ_U^- to be approximately equal. Similarly, one would expect the lighter side means, μ_L^+, μ_R^+, μ_U^+, to be close in value.

The photometric constraint requires more assumptions than the geometric constraints, and these assumptions are couched in approximate terms rather than exact equalities. The photometric constraint is therefore the weakest, yet it is still quite useful. If one assumes that,

- the surface on each side of the edge is indeed visible from all three cameras,
- the surfaces have roughly homogeneous values of reflectivity and surface normal away from the discontinuity,
- the observed intensity of a point does not change appreciably with camera location,

then one would expect to see a close match among the darker side means and the lighter side means.

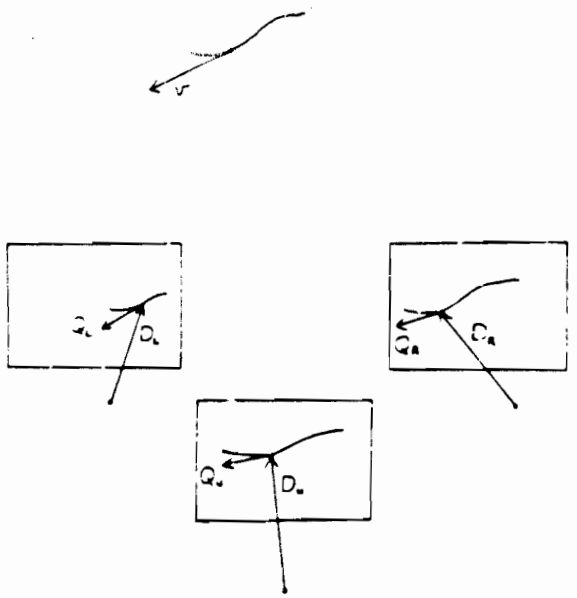


Figure 2-1: Camera Geometry for Edge Orientation

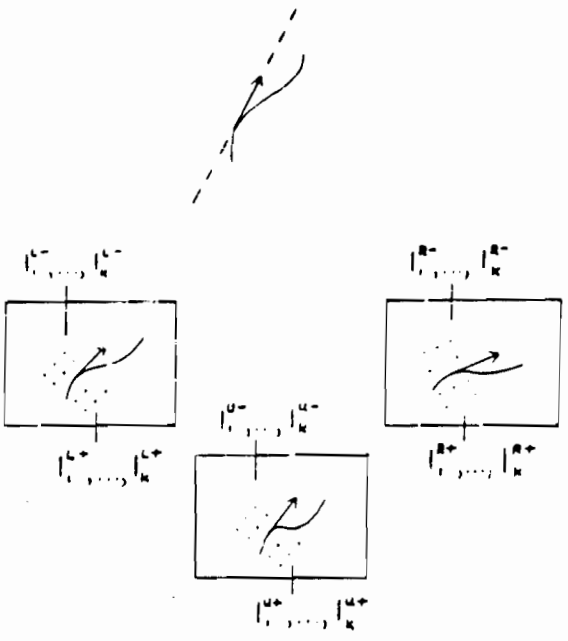


Figure 2-2: Sampling Images for Photometric Constraint

3. Statistical Matching

If the two geometric constraints and one photometric constraint held perfectly, then a matching algorithm could easily determine which candidate triples of edges were correct matches and which were not. Unfortunately, there are a number of sources of error that can cause a correct match to only approximately satisfy the geometric and photometric constraints, and these errors can also cause a non-matching triple to nearly satisfy these constraints and thus appear like a correct match. One error source is the physical limitations of the equipment such as finite digitization will can alter measured values. Occluding contours, if they arise from curved surfaces, and specular reflection create features that depend on the camera position and are therefore not matchable using the methods of this paper. The photometric constraint is based on a simple lighting and surface reflectance model. Cases where these simplifying assumptions do not hold can have moderate to large deviations from expected intensity values.

Even given these limitations, we can still maximize the number of correct matches by relying on a statistical algorithm. The following steps are necessary to generate such an algorithm:

- Assume the errors satisfy a known statistical distribution.
- Derive *confidence* measures that can be applied to evaluate hypothesized matches and to choose the best of a group of competing matches.
- Derive also *failure thresholds* which indicate which values of the confidence measure are too low to correspond to correct matches.

In case of multiple candidate matches for a feature, the algorithm chooses the one with the best confidence measure. Then it compares this measure against a failure threshold to determine if it should select this candidate as a correct match.

Sections 3.1 and 3.2 describe the general method for the derivation of confidence measures and failure thresholds. The following sections 3.3 and 3.4 describe the specific confidence measures and failure thresholds used for the edge orientation constraint and the local image intensity constraint.

3.1 Confidence Measures

In order to describe the derivation of the statistical confidence measure, let us use the following notation:

- The set of properties on the object will be denoted by P. For the edge orientation constraint, the relevant property will be the position of the contour point on the object and the tangent line to the contour at that point. For the photometric constraint, the property will be the light intensity reflected from a small patch of object surface.
- The set of values of observed features will be denoted by F (actually F^I, F^R, F^U for the three images). These feature values will be either edge orientation or image intensity.

In the first place, the set of object properties $p \in P$ must satisfy some statistical distribution $D_P(p)$. For example, we may know that the tangent line orientations are often distributed in the vertical and horizontal directions because of gravity. Secondly,

we must know the distribution of possible image feature values, $f^L \in F^L$ in the case of the left image, which can result as the projection of a given object property $p \in P$. In the ideal case, there will be no error in the imaging process and thus only one feature value f^L will correspond to an object value p. In general, however, errors and noise will spread the observed values into a distribution $D_{PF}(p,f^L)$. The notation is similar for the right and up images.

By combining the distribution of object properties with the distributions of feature values in each image corresponding to a given property value, we can derive the distribution for the set of property values and corresponding feature values,

$$D_{PF}^{LRU}(p,f^L,f^R,f^U) = D_P(p)\,D_{PF}^{L}(p,f^L)\,D_{PF}^{R}(p,f^R)\,D_{PF}^{U}(p,f^U).$$

This combined distribution is not useful because we cannot know the value of the object property p, and we can only directly observe the values f^L,f^R,f^L. Thus we integrate over $p \in P$ to obtain the distribution of observed feature values,

$$D_F^{LRU}(f^L,f^R,f^U) = \int_{p \in P} D_{PF}^{LRU}(p,f^L,f^R,f^U).$$

This likelyhood of observed feature values is the confidence measure we are seeking.

Suppose that for a feature in the left image with value f^L, we have two hypothesized matches A and B with feature value triples,

$$\langle f^L,f_A^R,f_A^U\rangle \text{ and } \langle f^L,f_B^R,f_B^U\rangle.$$

Let,

$$\alpha = D_F^{LRU}(f^L,f_A^R,f_A^U),$$

$$\beta = D_F^{LRU}(f^L,f_B^R,f_B^U).$$

Given that we must choose between A and B as the correct match, if $\alpha > \beta$, we should choose A because, on average, $\alpha/(\alpha+\beta)$ fraction of the time, it will indeed be the correct match. Thus we will make the correct choice that fraction of the time.

3.2 Failure Threshold

So far the analysis has shown us how to choose between competing candidate matches. We would also like to know if the match with the best confidence value is actually likely to be correct. Some sort of failure threshold for the confidence value is necessary. By definition an incorrectly hypothesized match corresponds to three features which are generated by different parts of the object and are therefore statistically independent. The distribution of feature values f^I generated independently is,

$$D_{IND}^{I}(f^I) = \int_{p \in P} D_P(p)\,D_{PF}^{I}(p,f^I).$$

If the feature values are statistically independent, the distribution for an triple $\langle f^L,f^R,f^U\rangle$ is the product of the individual distributions,

$$D_{IND}^{LRU}(f^L,f^R,f^U) = D_{IND}^{L}(f^L)\,D_{IND}^{R}(f^R)\,D_{IND}^{U}(f^U).$$

Suppose correct matches constitute fraction p_{true} of the set of hypothesized matches, and suppose false matches constitute fraction p_{false},

$$p_{true} + p_{false} = 1.$$

Let,

$$T = p_{true}\,D_F^{LRU}(f^L,f^R,f^U),$$

$$F = p_{false}\,D_{IND}^{LRU}(f^L,f^R,f^U).$$

If $T > F$, then the feature triple is more likely to have arisen from a single object property. If $T < F$, the features values are more likely to have arisen independently. Thus in order to select correct matches, we should compare the confidence measure of a match, $D_F^{LRU}(f^L,f^R,f^U)$, to a failure threshold,

$$\frac{p_{false}}{p_{true}}\,D_{IND}^{LRU}(f^L,f^R,f^U).$$

3.3 Edge Orientation Constraint

For the case of the geometric edge orientation constraint, the feature values are the angles $\langle \theta_L\ \theta_R\ \theta_U \rangle$ of the edge pixels in the three images. The object property is the orientation of the tangent line to the contour curve point on the object which generates the edge pixels. The value of the property is expressed as a point v on the Gaussian sphere. Let D_L be a vector which points along the ray from the camera center to the edge pixel and corresponding contour curve point as in figure 2-1. Let,

$$Q_L = (\cos\theta_L, \sin\theta_L, 0),$$

be a vector in the image plane aligned with the orientation of the edge pixel (Q_L and θ_L can be used interchangeably). The vectors D_L, Q_L, and v must lie in a plane, and therefore,

$$[D_L\ Q_L\ v] = 0 \quad \text{(triple product)},$$

$$Q_L = z\times(D_L\times Q_L), \quad \text{where} \quad z = (0,0,1).$$

The tangent line orientation v determines Q_L because Q_L is also constrained to lie in the image plane. In general the values of both Q_L and Q_R are necessary and sufficient to determine v,

$$V = (D_L\times Q_L)\times(D_R\times Q_R)$$

$$v = V/|V|.$$

These relationships hold exactly only in the ideal case, but whatever form the distribution $D_{PF}^{L}(v,\theta_L)$ (the probability the the image of v in the left image has orientation θ_L) takes, it will have a peak at the value of θ_L that corresponds to,

$$Q_L = z\times(D_L\times v).$$

3.3.1 Ideal Two Camera Distribution

Let us assume that $D_P(v)$ is a uniform distribution on the Gaussian sphere. In other words, all contour tangent line directions are equally likely. In the ideal case, the distribution $D_{PF}^{L}(v,Q_L)$ is a delta function about,

$$Q_L = z \times (D_L \times Q_L).$$

and each v projects to unique values of Q_I, Q_R, and Q_L in the three images. Nevertheless, as has been shown in [2], the set of pairs $\langle Q_I, Q_R \rangle$ do not form a uniform distribution even under these ideal circumstances. It is much more likely that matching edges in the right and the left images have the same orientation than it is that they have widely different orientations. The following is a derivation of a formula for this distribution $D^{LR}_{PF}(Q_I, Q_R)$ which is simpler and more efficient to compute than that given in [2]. The value of the distribution is the ratio between the area of a small region about v on the Gaussian sphere to the area of the projection of that region into the space of orientation pairs, $\langle \theta_I, \theta_R \rangle$. This ratio is the area spanned by v under infinitesimal changes in θ_I and θ_R as in figure 3-1.

$$V = (D_I \times Q_I) \times (D_L \times Q_L) \quad (1).$$

$$v = \frac{V}{|V|} \quad \text{(unit direction of tangent)}.$$

$$D^{LR}_{PF}(Q_I, Q_R) = \frac{1}{4\pi} \left| v \times \frac{dv}{d\theta_L} \times \frac{dv}{d\theta_R} \right|.$$

The factor of $(4\pi)^{-1}$ normalizes the result so that the integral of the distribution over all pairs of angles is unity.

Because v is of unit length and is perpendicular to $dv/d\theta_I$ and $dv/d\theta_R$, the area of the parallelogram formed by $dv/d\theta_I$ and $dv/d\theta_R$ equals the volume of the parallelepiped formed by v, $dv/d\theta_I$ and $dv/d\theta_R$. The volume of a parallelopiped can be expressed as a triple product,

$$D^{LR}_{PF}(Q_L, Q_R) = \frac{1}{4\pi} \left| v \cdot \frac{dv}{d\theta_L} \cdot \frac{dv}{d\theta_R} \right|.$$

Taking derivatives and substituting into this formula,

$$\frac{dv}{d\theta_L} = |V|^{-1} \cdot \frac{dV}{d\theta_L} + V \frac{d|V|^{-1}}{d\theta_L},$$

$$\frac{dv}{d\theta_R} = |V|^{-1} \cdot \frac{dV}{d\theta_R} + V \cdot \frac{d|V|^{-1}}{d\theta_R},$$

$$\left| v \cdot \frac{dv}{d\theta_L} \cdot \frac{dv}{d\theta_R} \right| = |V|^{-3} \left| V \cdot \frac{dV}{d\theta_L} \cdot \frac{dV}{d\theta_R} \right|.$$

Therefore,

$$D^{LR}_{PF}(Q_L, Q_R) = \frac{1}{4\pi |V|^3} \left| V \cdot \frac{dV}{d\theta_L} \cdot \frac{dV}{d\theta_R} \right| \quad (2),$$

where,

$$\frac{dV}{d\theta_L} = (D_I \times \frac{dQ_I}{d\theta_L}) \times (D_L \times Q_L) \quad (3),$$

$$\frac{dV}{d\theta_R} = (D_R \times Q_R) \times (D_R \times \frac{dQ_R}{d\theta_R}) \quad (4).$$

Thus given values for θ_I and θ_I, one can obtain V, $dV/d\theta_I$, and $dV/d\theta_R$, using equations (1), (3), and (4). From these vectors, equation (2) gives the value of the distribution.

Actually, Arnold [2] uses the one dimensional distribution of values of Q_R for a given value of Q_I, instead of the two dimensional distribution on $\langle Q_I, Q_R \rangle$. This one-dimensional distribution has a similar formula and derivation:

$$D^{LR}_{FF}(Q_I, Q_R) = \frac{1}{2\pi} \left| \frac{dv}{d\theta_R} \right| = \frac{1}{2\pi} \left| v \times \frac{dv}{d\theta_R} \right|,$$

$$D^{LR}_{FF}(Q_L, Q_R) = \frac{1}{2\pi} \left| \frac{V}{|V|} \times (|V|^{-1} \cdot \frac{dV}{d\theta_R} + V \frac{d|V|^{-1}}{d\theta_R}) \right|,$$

$$D^{LR}_{FF}(Q_I, Q_R) = \frac{1}{2\pi |V|^2} \left| V \times \frac{dV}{d\theta_R} \right|.$$

As can be seen in figures 3-2 and 3-3, these distributions have roughly the same shape, each favoring equal orientations in the two images. The two axes in these figures correspond to θ_L and θ_R ranging from 0 to 180 degrees in five degree increments.

3.3.2 Non-Ideal Three Camera Distribution

In general, the projections of the the contour tangent line will include some error, and so $D^I_{PF}(v, Q_I)$ will not be a delta function. Yet we know that if the edge pixel orientations have a high probability of being similar in the ideal case, they are likely to be similar even after they are perturbed by some error. When the distributions $D^I_{PF}(v, \theta_I)$, $D^R_{PF}(v, \theta_R)$, and $D^U_{PF}(v, \theta_U)$ are uniform and bounded, it is in fact possible to numerically compute the integral,

$$D^{LRU}_{F}(\theta_L, \theta_R, \theta_U) = \int_{v \in P} D_P(v) D^I_{PF}(v, \theta_L) D^R_{PF}(v, \theta_R) D^U_{PF}(v, \theta_U),$$

which is the likelyhood of a the triple of angles, $\langle \theta_L, \theta_R, \theta_U \rangle$. Figure 3-4 shows a slice through this three dimensional distribution for θ_I equal to 45 degree, for three cameras arranged in an equilateral triangle with unit side, and for a centered object point ten units away from the cameras. The measured orientation in each image is assumed to lie within four degrees of the ideal value with all values within the range being equally likely. The two axes correspond to θ_R and θ_L ranging from 0 to 180 degrees in five degree increments.

This three dimensional distribution takes too long to compute to be practically useful for a matching algorithm. It has, however, the following properties which make an approximation possible:

- The medial axis curve is the set of θ_R, θ_U which satisfy the error free case.
- The value of the distribution on the axis curve is roughly equal in value to the two dimensional distribution at the same value of θ_L and θ_R.
- Away from the axis curve, the distribution drops roughly linearly to zero in a distance of about eight degrees (twice the maximum angular deviation in each image).

Using these facts, one can numerically approximate the correct confidence measure using the formula for the two angle distribution derived above.

3.3.3 Failure Threshold for the Edge Orientation Confidence Measure

For the edge orientation constraint, it is relatively easy to compute the failure threshold. For edge pixels in one image, there is no reason to favor any edge orientation over another. Therefore the distribution is uniform on the unit circle. The distribution for a triple of independently generated edge pixels is simply the product of three uniform distributions,

$$D_{IND}^{LRU}(\theta_L,\theta_R,\theta_U) \equiv \frac{p_{false}}{8\pi^3 p_{true}}$$

For any reasonable values of p_{true} and p_{false}, this threshold cuts off the three dimensional distribution in figure 3.4 at a deviation of twice the maximum angular error in each image, in this case eight degrees.

3.4 Photometric Constraint

In section 2.3 we saw how the photometric constraint involves attempting to match two sets of three regions. One set arises from the low intensity region to one side of the edge pixels in the three images. The other set arises from the high intensity side. Sample values are taken in each region to generate the mean intensity values, $\mu_L, \mu_R, \mu_U, \mu_L, \mu_R, \mu_U$. The claim is made that if the three edge pixels arise from the same object contour point, the three low values should be related and the three high values should be related.

In sections 3.4.1 and 3.4.2, we will derive the confidence measure and failure threshold for matched regions in three images with k sample points each. In section 3.4.3, we will see how applying these results to the triple of darker side regions and to the triple of lighter side regions of a set of matched edge pixels enables us to detect edges that arise from occluding contours.

3.4.1 Confidence Measure of Photometric Constraint

The ideal case of matching three views of the same region (either three darker side regions or three lighter side regions) involves two assumptions. The first assumption is that the lighting is strictly Lambertian so that any point on the object appears with the same intensity from any viewpoint. The second assumption is that the entire region being sampled has constant reflectivity. Because the model we are using is so simple, any deviation from these ideal assumptions must be treated as noise. We will assume that the noise conforms to a Gaussian distribution.

Interestingly, these assumptions imply that we will see a greater variance among sample points taken from three views of the same region then we will see among sample points taken from one view of a region. The reason for this difference is that sample points taken from one viewpoint all see the same deviation from the ideal Lambertian lighting model. Therefore only the variance caused by non-uniform reflectivity affects them. Sample points taken from all three images are affected by both deviation from uniform reflectivity and deviation from the ideal lighting model and thus have a larger variance. Let us denote the one view variance by σ_1 and the three view variance by σ_3.

For the three image case, the distribution of image values I for a given object reflectance intensity x is,

$$D_{PF}(x,I) = (2\pi\sigma_3)^{-\frac{1}{2}} \exp\left(-\frac{(x-I)^2}{2\sigma_3}\right).$$

The combined distribution for the $3k$ sample points is,

$$D_{PF}(x,I_1^L,\ldots,I_k^L,I_1^R,\ldots,I_k^R,I_1^U,\ldots,I_k^U) =$$

$$(2\pi\sigma_3)^{-\frac{3k}{2}} \exp\left(-\frac{1}{2\sigma_3}\left(\sum_{i=1}^{k}(x-I_i^L)^2+\sum_{i=1}^{k}(x-I_i^R)^2+\sum_{i=1}^{k}(x-I_i^U)^2\right)\right).$$

Integrating over all values of x and taking twice the negative logarithm gives a confidence measure,

$$\frac{1}{\sigma_3}\left(\sum_{i=1}^{k}(\mu-I_i^L)^2+\sum_{i=1}^{k}(\mu-I_i^R)^2+\sum_{i=1}^{k}(\mu-I_i^U)^2\right) + (3k-1)\log\sigma_3 + C,$$

where,

$$\mu = \frac{1}{3k}\left(\sum_{i=1}^{k}I_i^L+\sum_{i=1}^{k}I_i^R+\sum_{i=1}^{k}I_i^U\right).$$

This confidence measure can be rewritten as,

$$\frac{k}{\sigma_3}VAR + \frac{k}{\sigma_3}(\sigma_L+\sigma_R+\sigma_U) + (3k-1)\log\sigma_3 + C \quad (5),$$

where,

$$VAR = (\mu-\mu_L)^2 + (\mu-\mu_R)^2 + (\mu-\mu_U)^2,$$

$$\mu_L = \frac{1}{k}\sum_{i=1}^{k}I_i^L, \quad \mu_R = \frac{1}{k}\sum_{i=1}^{k}I_i^R, \quad \mu_U = \frac{1}{k}\sum_{i=1}^{k}I_i^U,$$

and,

$$\sigma_L = \frac{1}{k}\sum_{i=1}^{k}(I_i^L-\mu_L)^2,$$

$$\sigma_R = \frac{1}{k}\sum_{i=1}^{k}(I_i^R-\mu_R)^2,$$

$$\sigma_U = \frac{1}{k}\sum_{i=1}^{k}(I_i^U-\mu_U)^2.$$

The confidence measure (5) consists of the sum of three parts. The first is proportional to the variance among the three means obtained from the three regions; the second part is proportional the sum of the variances seen in each region; and the third depends only on the choice of the three view variance σ_3 which is the same for all matches. Thus a good match will have both low variances in the three regions and closely matching means.

3.4.2 Failure Threshold for the Photometric Constraint

As has been mentioned above, we expect to see a lower variance among sample points taken from a single image. To calculate the failure threshold, we must first determine the probability density for k sample points under this smaller variance σ_1. Then we must take the product of densities for the three groups of k sample points. It turns out that twice the negative logarithm of this product is,

$$\frac{k}{\sigma_1}(\sigma_L+\sigma_R+\sigma_U) + (3k-3)\log\sigma_1 + C'$$

Hence, incorrect hypotheses can be detected by comparing the value of VAR, the variance among the means, against the threshold,

$$\frac{\sigma_3-\sigma_1}{\sigma_1}(\sigma_L+\sigma_R+\sigma_U) + C''(\sigma_1,\sigma_3),$$

a linear function of the sum of the variances in the individual images.

3.4.3 Detection of Occluding Edges

The failure threshold can enable the matching algorithm to determine if a triple of edge pixels has been generated by an occluding contour. For each hypothesized match, there are two photometric confidence measures, one generated by the three darker regions on one side of the edge pixels and the other generated by the three lighter regions on the other side of the edge pixels. In general, the algorithm uses the sum of the two confidence measures as the overall measure of the goodness of the hypothesized match. If one and only one side falls below the value of failure threshold for that set of intensity samples, then the algorithm hypothesizes that the edge is occluding. Recall that the failure threshold is really a measure of the confidence that the three regions are generated independently. Hence the overall confidence for an occluding edge is the sum of the failure threshold and the confidence measure for the other (matching) side.

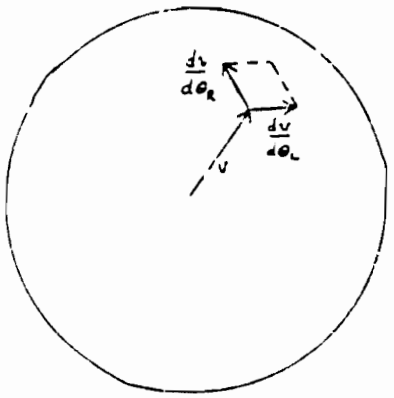


Figure 3-1: Area on the Gaussian Sphere about v Generated by Infinitesimal Changes in θ_L and θ_R

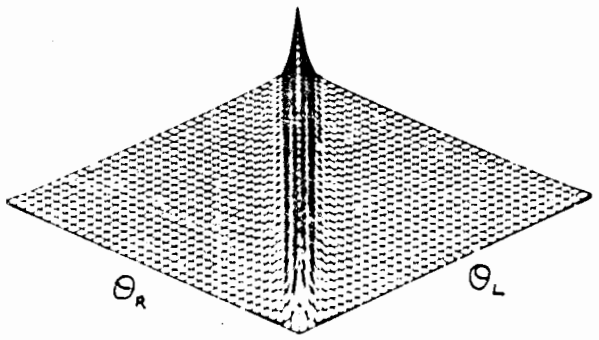


Figure 3-2: Distribution of $\langle \theta_L, \theta_R \rangle$

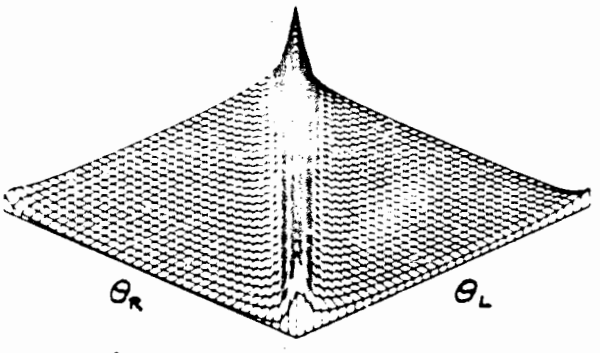


Figure 3-3: Distribution of θ_R as a Function of θ_L

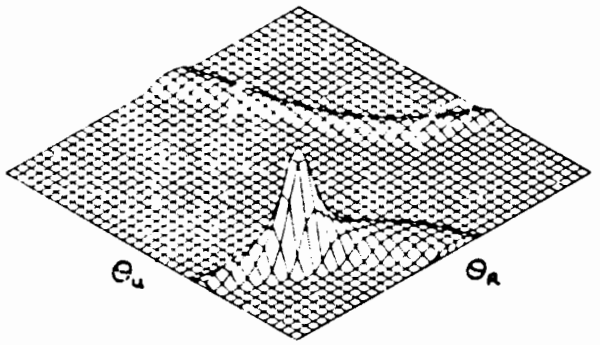


Figure 3-4: Slice through Distribution of $\langle \theta_L, \theta_R, \theta_U \rangle$

4. Filtering Competing Triples

We have developed two confidence measures for the trinocular matching algorithm to use in comparing competing matches. These measures imply a partial order on the likelyhood of matches. If one match has better confidence in both categories than another match, the first is clearly the more likely choice. Two matches are not necessarily orderable, however, if one has better edge orientation confidence and the other better photometric confidence. Despite this possibility, a trinocular stereo algorithm can obtain disparities for most of the edge pixels by means of the following steps:

- Generate all triples of edge pixels which satisfy the position constraint.
- Filter out matches for each pixel which are clearly less likely than another match at that pixel.
- Out of the set of unorderable match triples at each pixel, remove those which are not also strong matches at the other two pixels in the triple.
- If possible, cluster the remaining matches at each pixel by disparity.

The geometric position constraint on pixels is subject to some deviation, but not enough to make a statistical analysis of the error necessary. Generate the set of all matching triples as follows:

- Visit each edge point in the left image.
- For each point visited, visit all edge points in the right image within one pixel of the corresponding epipolar line.
- For each generated pair of edge points from the left and right images, check in the up image in a one pixel radius about the intersection of the corresponding epipolar lines. Every up image edge point found within this radius generates a match triple.

Suppose we have two triples of edge pixels which share the same edge pixel in the left image. If they are orderable, meaning one has better orientation confidence as well as photometric confidence, than the worst of the two is eliminated from the list of possible matches for that left edge pixel. It may also be beaten by triples with which it shares the same edge pixel in the right or up image. One hopes that the majority of triples will be rejected at all three locations and thus be dropped from consideration.

Once the algorithm has considered all matched triples, each edge pixel in each image will have a list of unorderable matches. Each triple in the list of a left image edge pixel may or may not have been rejected from the appropriate lists for the right and up image. Clearly it is better that a triple be accepted in all three images, less good if it has been rejected once, and worst yet if it appears in a list in only one image. Thus the algorithm has a new basis for comparison of the unorderable triples at a given edge pixel. It can reject those triples in the list which have been rejected more times than the best element.

At this point, the algorithm clusters unorderable triples in a given list by disparity distance. If all the triples agree to within a few pixels, than the algorithm takes the average disparity as the correct value. If the unorderable matches do not lie in a single cluster, no correct match can be made.

5. Performance

The trinocular edge matching algorithm was applied to a set of synthetic images and two sets of real images. In each case it performed well, despite the fact that it used no continuity principles (neighboring edge pixels should have the same depth) or other high level information. The algorithm also ran relatively rapidly, using no more than five minutes on a VAX/785 to process the real image.

5.1 Synthetic Image

A graphics system generated the synthetic image. It consists of a collection of overlapping "boxes" created from rectangular plates. Each plate has a smaller rectangular region in its center with a slightly lower reflectivity. The lighting model is Lambertian, with two sources of light, one behind the cameras to the upper left, and one behind the cameras to the upper right. The image was created at a resolution of 512 by 512 pixels and then reduced by averaging to 256 by 256 pixels. The cameras are arranged in an equilateral triangle, and the disparity ranges from 40 to 60 pixels. The three images are shown in figures 5-1, 5-2, and 5-3, and the edge images in figures 5-4, 5-5, 5-6.

In order to test the algorithm, this image was deliberated made cluttered with both occluded edges and occluding contours. It has, however, a number of advantages over a real image. The edge orientations and positions are very accurate, and the sample variances on a particular surface are very low. In other words, most of the assumptions necessary for good performance of the confidence measures hold true. No assumptions, however, were made about the range of disparities except that points in the right image were presumed to be to the left of points in the left image. The algorithm used no multiple resolution or other continuity techniques.

The algorithm was run twice in order to test the detection of occluding edges. In the first run, no failure threshold was used for the photometric confidence measure. In the second run, the failure threshold and the scheme for detecting occluding edges was used. Figure 5.2 shows the result of the matching where edge points with a larger calculated disparity are represented by darker, thicker lines. As expected, in the first run, the algorithm failed to match edges arising from occluding contours. In the second run with the failure threshold, it matched most of these without a significant degradation in performance elsewhere in the scene. Figure 5.2 shows the result of this second run, and figure 5.2 shows the correct disparity map for all unoccluded edge points (which could be calculated since these were synthetic images). In both cases, the performance was excellent on finding correct matches when they existed, although the algorithm did tend to accept an incorrect match when no correct match existed.

5.2 Real Images

Two sets of images were taken in the vision lab at CMU. The first set was generated by mounting the camera to the slider of a drawing table. By turning the table vertically, we constrained the camera motion to the horizontal and vertical directions. The three camera positions formed an equilateral triangle, but because the camera was held in a slightly rotated position in the mounting, the camera model had to be determined by hand matching 25 points in the three images. Instead of rectifying the images, the trinocular algorithm was modified to allow the epipolar line orientation to vary with pixel position. Unfortunately, due to the crudeness of the mounting arrangement, the model showed as much as two pixel deviation of edge points from its corresponding epipolar lines. The algorithm was modified to search in this range. Figures 5-10, 5-11 and 5-12 show the first set of images, and figures 5-13, 5-14 and 5-15 show the edge points. The second set of real images were generated using a tripod and marks on the floor. The camera model for this second set much more closely matched the ideal equilateral case. Figures 5-16, 5-17 and 5-18 show the first set of images, and figures 5-19, 5-20 and 5-21 show the edge points.

The images were 256 by 240 pixels (reduced by averaging from the 512 by 480 camera output) and the disparity range was 20 to 60 pixels in the first image and 18 to 55 pixels in the second image. The algorithm was restricted to search only in this disparity range. The photometric failure threshold was used.

The results were good for both scenes as shown in figures 5-22 and 5-23. In the first scene, the matching was better in the center of the image where the camera model was more accurate. In the second scene, the matching was good overall.

Figure 5-1: Synthetic: Left Image

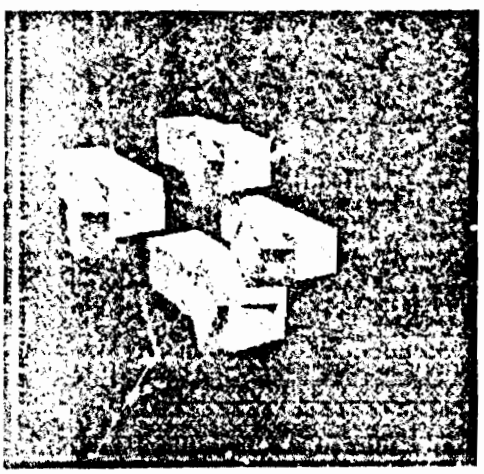


Figure 5-2: Synthetic: Right Image

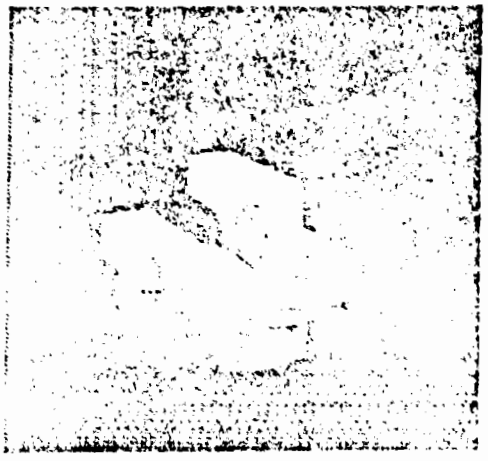


Figure 5-3: Synthetic: Up Image

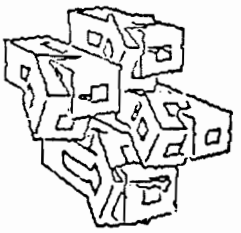


Figure 5-4: Synthetic: Left Edges

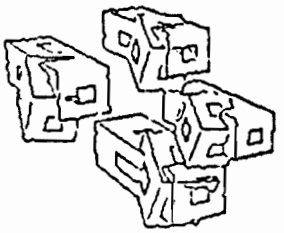


Figure 5-5: Synthetic: Right Edges

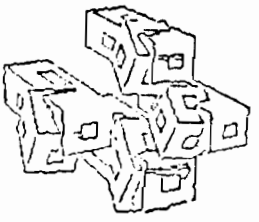


Figure 5-6: Synthetic: Up Edges

Figure 5-7: Matching without Failure Threshold

Figure 5-8: Matching with Failure Threshold

Figure 5-9: Correct Matches

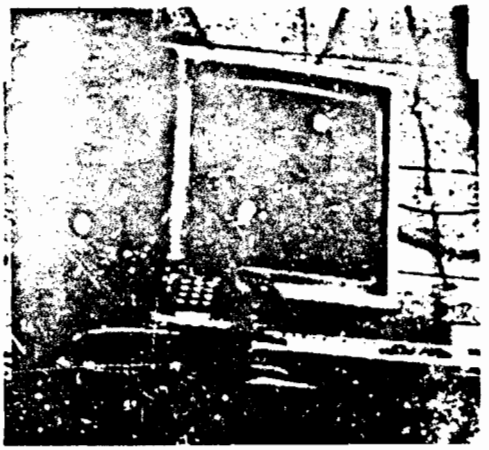


Figure 5-10: First Real: Left Image

Figure 5-11: First Real: Right Image

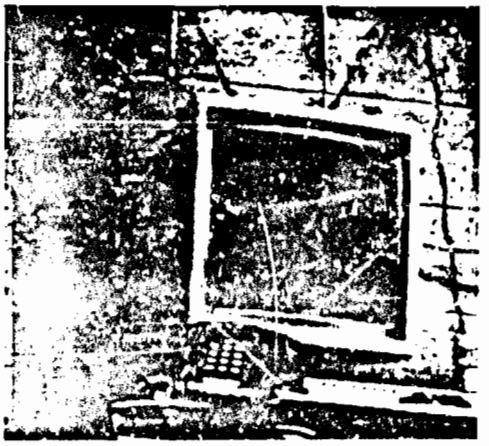


Figure 5-12: First Real: Up Image

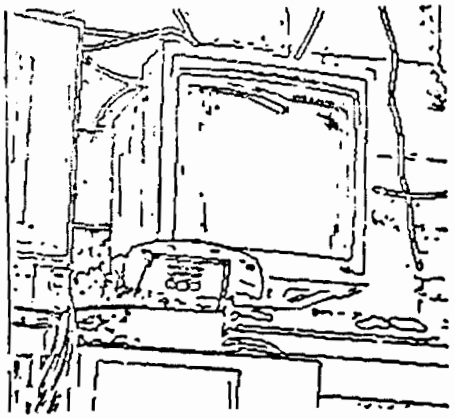


Figure 5-13: First Real: Left Edges

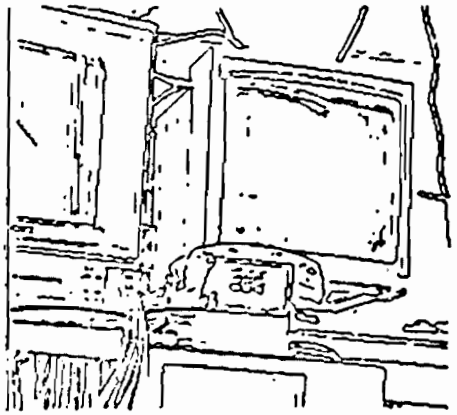


Figure 5-14: First Real: Right Edges

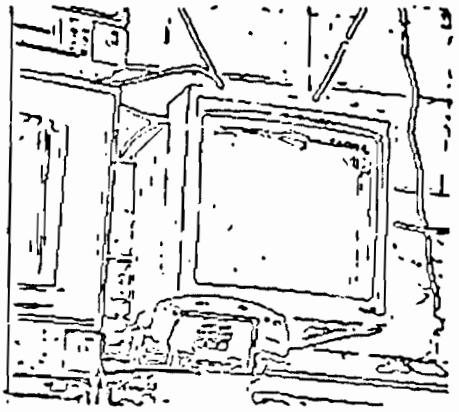


Figure 5-15: First Real: Up Edges

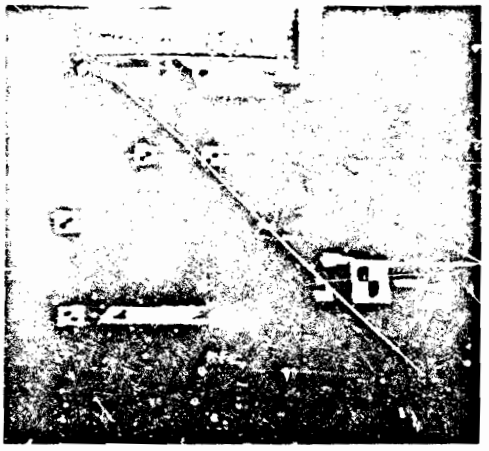


Figure 5-16: Second Real: Left Image

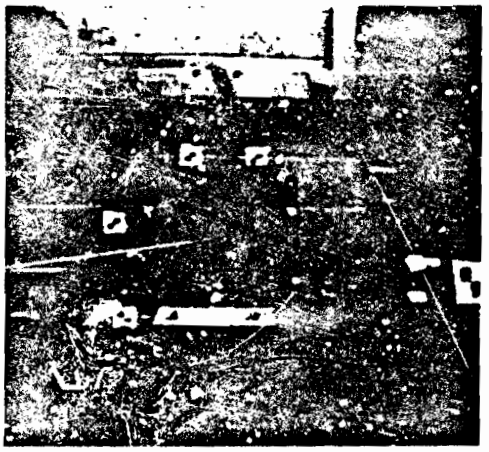


Figure 5-17: Second Real: Right Image

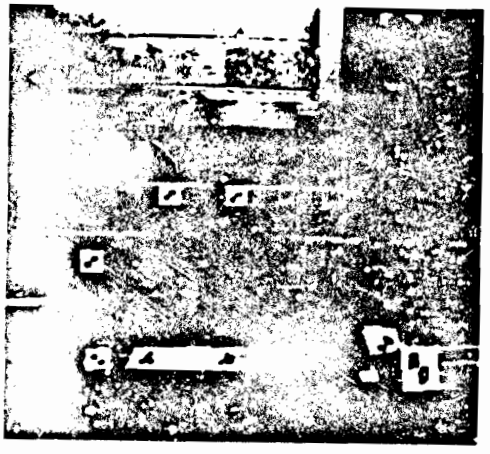


Figure 5-18: Second Real: Up Image

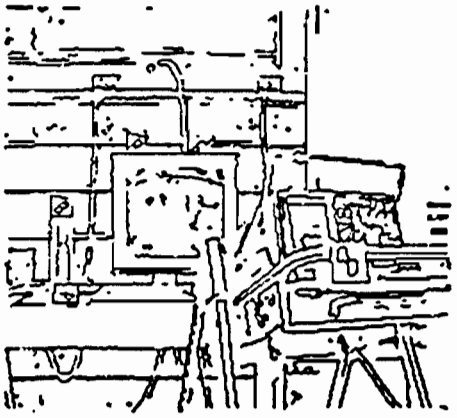


Figure 5-19: Second Real: Left Edges

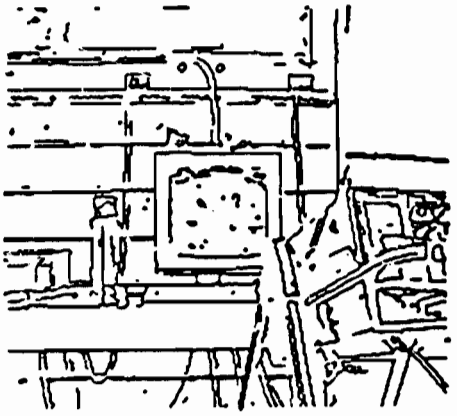


Figure 5-20: Second Real: Right Edges

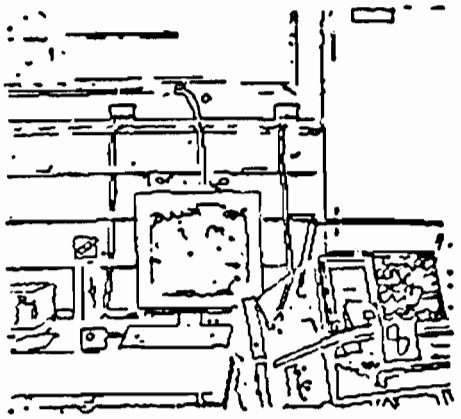


Figure 5-21: Second Real: Up Edges

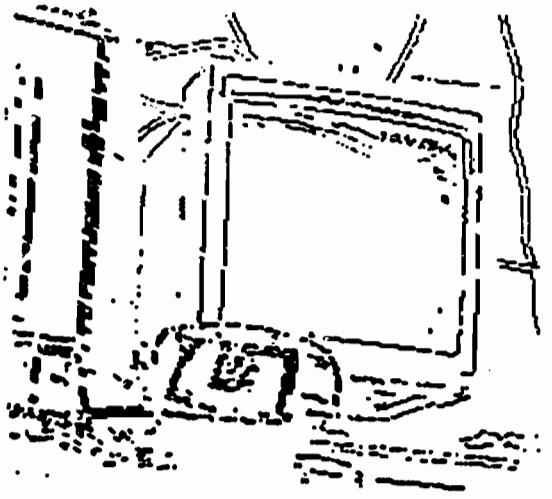


Figure 5-22: Depth Map for First Real Scene

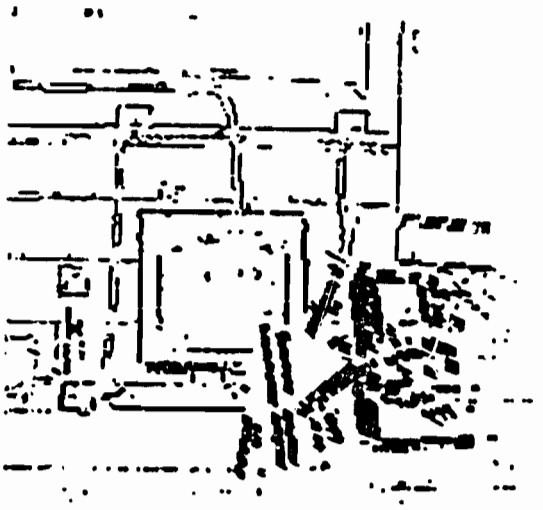


Figure 5-23: Depth Map for Second Real Scene

6. Conclusion

The trinocular matching algorithm performs very well even though it uses the same order of computing resources as the binocular method. And because of the additional constraints provided by the third image, the trinocular algorithm does not depend on any continuity assumptions which are usually necessary to make a binocular matching scheme work. The trinocular method can also match horizontal and near horizontal edges, which the binocular scheme cannot.

Both the binocular and trinocular methods depend rather strongly on an accurate camera model. An inaccurate camera model forces the algorithm to scan a strip around the epipolar constraints. The resulting increase in the number of geometrically matched triples to be evaluated and compared degrades the running time of the algorithm and increases the number of false matches and missed matches. The trinocular algorithm coped with a two pixel error in the case of the vision lab image but at some cost in performance.

The algorithm presented here is a true trinocular algorithm. It considers all three images at the same time, not as three binocular pairs. As a result, the number of occluded contour points is greater than for any of the binocular pairs, because in order to be matched, a contour point must be visible from all three cameras. If the algorithm were extended to use higher level assumptions, it could be made to match points visible in only two out of three images. For example, the system could perform three dimensional curve tracing in order to reconstruct the three dimensional contours on the object. The increased understanding of the object structure could enable the system to eliminate "single bit" errors. More important, once this step is taken for the contour points visible from all three images, the contours could be extended based on the information available in two of the images. Hence, even if some parts of a contour were not visible in one image, the contour points could still be matched reliably. In this case, the trinocular method would match more points than the binocular method because more points are visible from two out three cameras than from two out of two.

References

1. John Aloimonos, Amit Bandyopadhyay, and Paul Chou. On the Foundations of Trinocular Machine Vision. University of Rochester Department of Computer Science, Rochester, NY 14627, May, 1985.

2. R. D. Arnold and T. O. Binford. Geometric Constraints in Stereo Vision. Computer Science Department, Stanford University, 1980.

3. S. T. Barnard and W. B. Thompson. "Disparity Analysis of Images". *IEEE Transactions on Pattern Analysis and Machine Intelligence 2* (1980), 334-340.

4. Minoru Ito and Akira Ishii. Three-View Stereo Analysis. Musashino Electrical Communication Laboratory, Nippon Telegraph and Telephone Public Corporation, 3-9-11, Midori-cho, Musashino shi, Tokyo, 180 Japan, 1985.

5. J. Y. S. Luh and John A. Klaasen. "A Three-Dimensional Vision by Off-Shelf System with Multi-Cameras". *IEEE Transactions on Pattern Analysis and Machine Intelligence 7*, 1 (January 1985), 35-45.

6. Y. Ohta and T. Kanade. Stereo by Two-Level Dynamic Programming. Proceedings of the Ninth IJCAI, 1985, pp. 1120.

7. Roger Y. Tsai. Multiframe Image Point Matching and 3-D Surface Reconstruction. Research Report RC 9398 (# 41469), IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, May, 1982.

8. Masahiko Yachida. 3-D Data Acquisition by Multiple Views. Third International Symposium on Robotics Research, Paris, France, October, 1985.

CC970145-23

Edgel-Aggregation and Edge-Description

Vishvjit S. Nalwa
A.I. Lab., Stanford University, CA 94305

Eric Pauchon†
E.T.C.A., 94114 Arceuil Cedex, France

Abstract

An edge in an image corresponds to a discontinuity in the intensity surface of the underlying scene. It can be approximated by a piecewise straight curve composed of edgels, i.e. short, linear edge-elements, each characterized by a direction and a position. In a previous paper [Nalwa'84] we described a strategy to detect edgels. Edgels, by themselves, are of little use in vision systems. In this paper we proceed to discuss algorithms to aggregate edgels into edges and to describe these edges by best-fit curves.

The edgel-linking algorithm is simple and has a local character. It relies only on edgel proximity and direction. The basis used for edge-description consists of conic-sections. Position and tangent continuity are maintained in the curve-fitting stage. The problems addressed include, the discovery of straight lines and their discrimination from low-curvature segments, the detection of corners, the choice of knots and the estimation of the distance of an edgel from a conic-section. We demonstrate our algorithms with a detailed example.

1. Introduction

It is hard to over-emphasize the importance of edge-detection in image understanding. Most modules in a conceivable vision system depend, directly or indirectly, on the performance of the edge-detector. Consequently, there has been a substantial effort in this direction. Despite this effort, many in the community believe that the problem is largely unsolved. In fact, it may be claimed with some justification, that research and motivation on other fronts (e.g. stereo and line-drawing interpretation) has been dampened by the ineffectiveness of existing detectors.

An edge in an image corresponds to a discontinuity in the intensity surface of the underlying scene. It can be approximated by a piecewise straight curve composed of edgels, i.e. short, linear edge-elements, each characterized by a direction and a position. In a previous paper [Nalwa'84] we described a strategy to detect edgels with step-profiles, which by far are the most common type. Edgels, by themselves, are of little use in vision systems. In this paper we proceed to discuss algorithms to aggregate edgels into edges and to describe these edges by best-fit curves.

The problem addressed in this paper is posed as the following. Given, a list of edgels belonging to an image, where each edgel is characterized by its orientation, the location of its center, the intensities on its two sides and the window in which it was detected. First, aggregate the edgels into ordered sets corresponding to individual extended edges. Then, describe these edges by fitting curves to their edgel-members in some best-fit sense. Position and tangent continuity are to be preserved in the curve-fitting stage. Also, the unique role played by straight lines in subsequent vision-system-modules demands that we distinguish them from low-curvature segments.

Compared to the vast literature on edge-detection, the work on edgel-linking is meager. A review of previous work is to be found in [Ballard & Brown'82]. Our approach to edgel-linking has a local-character and employs few heuristics or thresholds. In fact, it is simple enough for us to believe that it is likely to have been implemented previously, although we have been unable to find any published source. Our algorithm is a three step process. Step 1: map the edgels onto a grid. The spacing of the grid is determined by the quality of the edgel-detector. We use a square grid with half-pixel spacing. Step 2: thin the edgels mapped onto the grid so as to obtain minimum connectivity, in the 8-neighbour sense, between the edgel-centers. This thinning stage must be distinguished from the common thinning procedure used on the output of many edge detectors. The aim of Steps 1 and 2 is not to localize the edge, but only to avoid all search in Step 3 by forming a minimally connected graph between the edgels. Step 3: starting with a ungrouped edgel, extend the edgel-set in both directions by following the connectivity-graph obtained from Step 2. Contour-following rather than search is involved in this stage. Decisions about the choice of the next member of the the current edgel-set are to be made only at junctions and are based on local orientation compatibility. The details of this algorithm are discussed in Section II.

This work was supported in part by the Defense Advanced Research Projects Agency under contract N00039-84-C-0211.

†EP was a visiting research scientist at the Stanford A.I. Lab. during the 1984-85 academic year.

Similar to the approach in [Turner'74], we use the plot of tangent vs arc-length to segment our edges into conic-sections and straight lines. The local orientation along the curve is directly available to us from the edgel parameters and the arc-length is estimated by obtaining a polygonal approximation to the edge. The intrinsic representation is used to discover straight lines and to choose candidate knots from among the edgels. It is also used to detect corners. We use conic-sections to describe non-straight edge segments. Although conics has been often used to fit approximating curves to data [see Pavlidis'83], unlike most previous attempts we also deal with the problem of position and tangent continuity between adjacent segments. When one chooses to fit a curve based on an error criterion dependent on the distance of the data from the curve, an important accompanying concern is the estimation of this distance. It is often a non-trivial problem and crude approximations show up in the fitting-curves having systematic deviations from the optimum fit. We formulate a distance measure, for a point from a conic, which overcomes some of the problems associated with previous formulations. We also indicate how one could obtain an exact solution for this distance and why it is impractical to use it. The details of the curve-fitting approach are outlined in Section III.

We should mention that it is not our intention to give detailed algorithms in this paper. Only broad outlines are presented and some of the issues we have concerned ourselves with, are discussed. The details of the specific implementation are numerous and may vary.

In Section IV, we present and discuss a detailed example and finally in Section V we conclude with an outline of some of the important issues in linking and curve-fitting which need further research.

II. Linking

As mentioned in the introduction, our linking algorithm is local-based and is essentially a three-step process. Our chief concern in its design was simplicity and effectiveness without unnecessary heuristics.

In the first step, the edgels are mapped onto a square grid which we call the connectivity-grid for reasons which will soon become clear. This grid is conceived to be a 2-D array of cells. Each cell contains a flag and a pointer. The corresponding data-structure is sketched in Fig. 1. The grid-spacing should be chosen on the basis of the quality of the edgel-input. The spacing should be such, that the resolution of the edgel-detector is maintained and edgels belonging to the same edge not have gaps between their mappings. We chose a half-pixel spacing for edgel-data obtained from the application of the Nalwa Operator [Nalwa'84] to an image. Some of the masks used for the mapping are shown in Fig. 2. The appropriate portion of the corresponding mask is copied onto the grid. The flag of a cell indicates whether an edgel-center, an edgel-extension or the detecting-window or none of the above is mapped

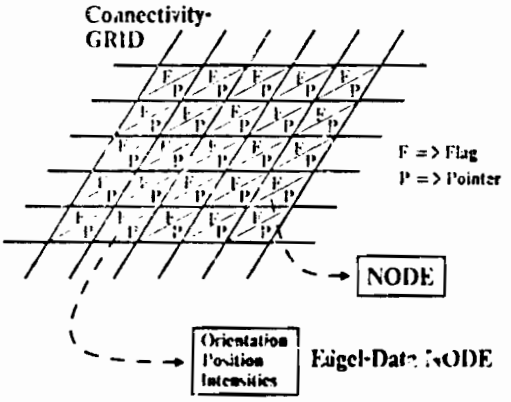


Fig. 1. Connectivity-Grid.

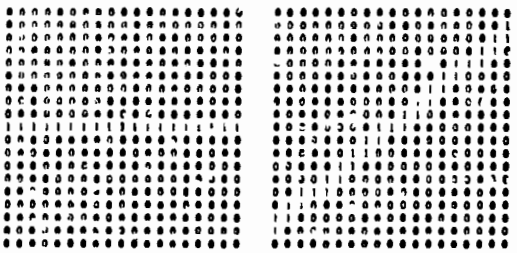


Fig. 2. Sample masks used to map edgels onto the connectivity-grid.

onto the cell. An edgel-center flag supersedes an edgel-extension flag. The pointer in each cell with an edgel-center flag points to a node containing the corresponding edgel-parameters, i.e. position, orientation and intensities. If more than one edgel-center is mapped onto a cell then the various parameters are averaged. We must emphasize that the connectivity grid is used only to obtain connectivity information for the edgels.

As some of the edgels will inevitably be tangent to high-curvature edges, their extremes will deviate from the underlying edge and may result in false connectivity during the first step. Hence, it may be desirable to disregard the non-overlapping extremes of edgels.

In the second step, we thin the edgel mappings onto the connectivity-grid in order to obtain a minimally connected (in the 8-neighbour sense) graph. For this purpose, one can use any of the standard thinning algorithms [see Pavlidis'82] with minor modifications. We use what Pavlidis [Pavlidis'82] calls the Classical Thinning Algorithm, simply because it was the easiest to implement. We first thin out all the non-skeletal edgel-extension cells. Then, we thin the grid so as to obtain a minimally connected graph (in the 8-neighbour sense). During this process care is

taken not to discard any edgel-center cell without doing one of the following. If it neighbours a skeletal edgel-center cell, the parameters are averaged appropriately, else a pointer from a neighbouring skeletal edgel-extension cell is established to its data-node. Essentially, we do not want to carelessly discard any information. By reducing the connectivity grid to a minimally connected graph, we ensure that when we group edgels by contour following in step three, we will obtain a nearly completely ordered set, i.e. the edgels will be ordered on the basis of their position along the edge. This will be important for our curve-fitting strategy.

In the third step we group the edgels into ordered sets, each corresponding to an edge. This stage simply involves starting out with an unclassified edgel-center cell and extending the edgel-set by following the minimally connected graph in both directions. The starting edgel-center is chosen to be at most 2-connected in order to avoid the complications of starting out at a junction. Decisions about the choice of the next member of the set arise only at cells with connectivity greater than 2, i.e. at junctions. These decisions are based on the compatibility of the orientation of the last edgel-member of the current set and that of the candidate edgels.

It is often feasible to extend edges and connect adjacent edge-terminations if one lowers the threshold on the edgel-contrast. One way to incorporate this capability into the above algorithm is to obtain the connectivity graph for all edgels with a contrast greater than the lower threshold and then, in step three, to discard all edgel-sets which do not have any edgel with contrast greater than the higher threshold. However, we must be careful about random edge-extensions which do not have orientation-compatibility with the main-edge.

At the conclusion of the linking stage, we have ordered lists of edgels belonging to individual extended edges. Junction information is also maintained, i.e. information about edgel-list terminations at junctions and edgel-list intersections is preserved. We remind the reader that the relevant parameters, i.e. orientation, position and intensities, for all edgels remain accessible.

III. Curve-Fitting

Some of the issues encountered when attempting to fit curves to data, are the following. What should be the order of continuity between adjacent curve segments? It is well known that humans are sensitive to both, position and tangent discontinuities. Then, we must choose the family of curves we intend to fit. This choice must be compatible with our continuity requirements and must also be amenable to reasonable error-criteria formulations. Next, we must decide on how to choose knots, given our choice of curves. We may also want to discover instances of certain features (e.g. straight lines, corners) which play an important role in subsequent processing.

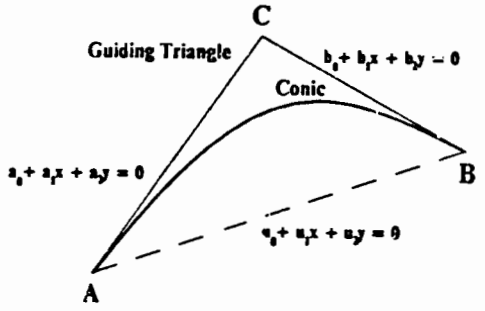


Fig. 3. Guided Conic.

Because humans are sensitive to C_1 discontinuities in curves and, in a sense, our confidence about the feasibility of machine vision is based on their performance, we believe that tangent continuity is a necessary requirement for fitting curves to edgel-data. As regards curvature discontinuities, we consider the empirical evidence in support of human sensitivity to them to be insufficient. Having decided on the order of continuity desired, we must now choose a family of curves. We decided to use conics for a variety of reasons. Firstly, they can satisfy our continuity requirements. Secondly, unlike with functions of an independent variable, we do not have to deal with awkward segmentation at points with infinite slope. Further, the family of conics have been studied as far back as Appollonius and their properties are well documented. Also, unlike higher order curves they do not have inflection points. The implications are that we must introduce knots at inflection points, but more importantly that if the knots are well-chosen then unnecessary wiggles are avoided.

An excellent introduction to conics is given in [Pavlidis'83]. It indicates how one may proceed to fit conic splines to data without solving explicitly for the six parameters in the algebraic representation $a.x^2 + b.x_y + c.y^2 + d.x + e.y + f = 0$. We repeat the formulation here. Let A and B be two points with known tangents $a_0 + a_1x + a_2y = 0$ and $b_0 + b_1x + b_2y = 0$ respectively, as shown in Fig. 3. Further, let the line connecting A and B have the equation $u_0 + u_1x + u_2y = 0$. Then, the family of conics which passes through A and B with their specific tangents has one degree of freedom, say K. It can easily be verified that $K.(u_0 + u_1x + u_2y)^2 = (a_0 + a_1x + a_2y)(b_0 + b_1x + b_2y)$ represents this family. We have the freedom to choose K.

Given our choice of conics and the C_1 continuity constraint, we have a one-parameter family of curves for every given set of edgels with their end-points and tangents at their end-points specified. We minimize the sum of square-errors to determine K, i.e. minimize Σd_i^2 where d_i is the distance of the i^{th} edgel from the conic. In Appendix I, we indicate how d may be estimated. Our estimation technique is more

accurate than previous approximations [e.g. Turner'74] and overcomes some of their drawbacks, e.g. a tendency to produce "flat" conics-fits. We indicate in Appendix II how one may proceed to find the exact distance of a point from a conic and why it is impractical to do so. It can immediately be seen that our error-criterion is invariant to equiform transformations of the image-plane, i.e. our best-fit curve undergoes the same translation, rotation and change in scale as the edgel-data. This, of course, is a very desirable feature.

Now we address the problem of segmentation of a set of edgels into sub-groups, each of which will be fit with a separate curve maintaining position and tangent continuity at the knots. As mentioned before, each edgel has its center-position and orientation specified. Also, the edgels in every set are ordered on the basis of their position along the contour. We use the ψ-s representation, i.e. the plot of tangent vs arc-length, to determine the knots. As the orientation of every edgel is known, we only need an estimate for the arc-length at the edgel-centers to obtain the ψ-s plot. It is not advisable to estimate s by simply taking the cumulative sum of the inter-edgel distances because of the "coast-line" effect, i.e. such an estimate will over-shoot the actual length owing to the scatter of the data about the underlying curve. We use a polygonal approximation to the edgels, to determine s. Our line-fitting algorithm is adapted from a split-and-merge algorithm listed in [Pavlidis'82]. Once a polygonal approximation to the data has been obtained, the distance of an edgel from the preceding vertex is used as an estimate for the distance between the two points, along the curve. In this fashion we construct the ψ-s plot for each set of edgels. The ψ-s curve has some interesting properties of which we would like to make the reader aware. Translation of a curve in the image-plane does not affect its ψ-s plot, rotation corresponds to a shift in the ψ-axis by the angle of rotation and change of scale corresponds to a proportional stretching or shrinking of the s-axis. Closed contours in the image plane have periodic ψ-s plots. Further, the slope of the ψ-s plot, i.e. $d\psi/ds$, gives the curvature of the corresponding point on the curve in the image-plane. It follows, that straight lines in the image-plane manifest themselves as zero-slope segments in the ψ-s plane, inflection points map onto extrema, and circles onto constant-slope segments.

Having obtained the ψ-s plot we first seek out straight lines in the image. We begin by thresholding on the curvature and length of segments of the polygonal-fit in the image-plane to obtain candidate straight-lines. The curvature is estimated from the slope of the best-fit (in the least-squares sense) linear approximation in the ψ-s plane. Ideally, straight lines must have zero curvature. Our threshold on maximum curvature was 1°/pixel. This corresponds to a circle of radius 60 pixels approximately. The purpose of the threshold on minimum length is to avoid fitting straight lines to low curvature segments of larger curves with varying curvature, e.g. portions of "flat" ellipses. Thresholding on curvature obviously does not distinguish between straight lines and curves with curvature less than the threshold, even though the orientation at the two end-points of the segment may be distinctly different. Hence, we also threshold on the difference in the orientation estimates (obtained from the linear-fit in the ψ-s plane) at the end-points. We also check if the constant-orientation hypothesis is satisfied in the ψ-s plane. Adjacent straight-line candidates must either merge into a single straight line or they must have a tangent discontinuity between them. Tangent discontinuities are checked by comparing the orientation estimates for the common datum from the two adjacent linear-fits in the ψ-s plane. Merging is successful if the curvature remains within bounds, the total change in orientation is within limits, the constant-orientation hypothesis is valid in the ψ-s plane and a linear-fit in the image plane is successful.

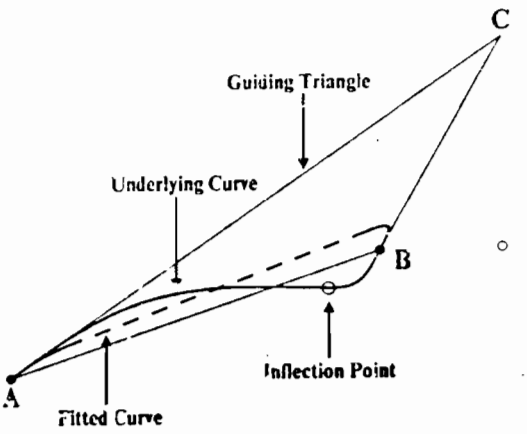


Fig. 4. Guided conic fitted to data with an internal inflection point.

Having found straight-lines, we now turn our attention to inflection points. Inflection points on a curve are points at which the curvature changes sign. Therefore, they appear as extrema in the ψ-s plot. It is our speculation that they play an important role in human vision. Unlike extrema in curvature, they are preserved under perspective projection. This is easily verified by noting that straight lines remain straight under perspective projection. We find extrema in the ψ-s plane by using the hysteresis smoothing mechanism [Duda & Hart'73]. It is a non-linear process for finding "significant" extrema.

Now we address the issue of knot placement. We begin by placing knots at the ends of straight lines and at inflection points. As mentioned previously inflection points cannot be represented by conic sections. If we try to fit a conic to data with an internal inflection point, the situation shown in Fig. 4 will result. In fact, we must explicitly check for this pathological case before we attempt conic-fitting because, inevitably, some inflection points would have escaped detection. To position the other knots we

obtain polygonal-fits to portions of the ψ–s plot between the knots already chosen. The algorithm used is once again of the split and merge variety from [Paviidis'82]. The corners of the polygonal-fit are used as an initial choice for the remaining knots. The segments between these knots correspond to data which can approximately be described by a circle, i.e. a constant-curvature curve, in the image-plane. Hence, we expect the family of conics to be sufficient to describe them. We have also, in an indirect fashion, achieved orientation compatibility between the edgel-data and the fitted conic in the image plane. The final number and placement of the knots, chosen from the polygonal-fit in the ψ–s plane, is determined by a split and merge algorithm which keeps the least mean-squares-error and the maximum unsigned error of the best-fit conic, in the image-plane, within bounds. As mentioned in Appendix I, our error-distance estimates are never smaller than the actual error. Hence, the error bounds are strictly satisfied, although this may be at the expense of extra knots. Systematic errors, due to the insufficiency of the basis, can be discouraged by modulating the error thresholds to favor crossings of the fitted curve and the underlying data, i.e. sign changes in the error-distances of ordered edgel-data are favored.

An often mentioned example in support of knot placement at curvature extrema is Attneave's cat [Attneave'54]. We believe, that the choice of knots must necessarily be guided by the choice of the family of curves one uses to represent the data. The fact that Attneave found that a cat's outline was still easily recognizable when it was represented by straight lines joining curvature-maxima only indicates that knot placement at curvature-extrema is a good idea if one is using a polygonal approximation. This is fairly intuitive if one considers the Taylor-Series expansion of a curve in the ψ–s plane. As mentioned before, a straight line in the image-plane corresponds to a constant in the ψ–s plane. Therefore, if we represent a curve in the image-plane by straight-line segments, it corresponds to a representation by constant segments in the ψ–s plane. From the Taylor-Series expansion for a function about a point, it follows that the error term for a function approximated by a constant depends on its first-derivative in the interval of approximation. Hence, we would like to locate the first-derivative extrema near the bounds of the approximating interval. But these extrema in the ψ–s plane are precisely the points of maximum curvature in the image-plane. By the same argument, it seems reasonable to locate the knots at maxima of $d^2\psi/ds^2$ for approximation by circular arcs in the image-plane. This stream of argument supports our strategy of choosing knots.

Besides discovering straight-lines and inflection-points, one may also want to detect tangent-discontinuities because of their special role in human vision. In all its generality, this is a hard problem. The finite size of the edgel-detection operator causes the failure of the straight-edge-segment hypothesis used by edge-detectors, near corners. Consequently, the orientations of edgels detected near corners are "blurred" versions of the actual orientation. This "blur," is over and above the relatively small amount of "blur" introduced by the imaging system. Because of the "smoothing" of corners, there is no way to distinguish between them and edge-segments with curvature comparable to that of "smoothed" corners without going back to the image. The problem becomes more acute as the size of the edge operator is increased. Keeping the above observations in view, we believe that an advisable strategy is to obtain candidate corners from the edgel data and then check the hypothesis in the image. We have not implemented the suggested approach. Instead, we have devised a simple procedure to detect high-curvature segments whose lengths are explicable as responses, for the given edge-detector, to corners, i.e. if the length of a high-curvature segment is comparable to the edge-operator-width, then it is a candidate corner. High curvature segments can be obtained by thresholding on the slope of the polygonal-fit segments in the ψ–s plane. To avoid responding to noise, a lower bound is placed on the total angle change between the ends of candidates segments. The declaration of corners at segments of curves with smoothly varying curvature, e.g. ends of "flat" ellipses, is discouraged by insisting on a significant curvature change between the candidate corner-segment and the adjoining regions. From the arguments above, it is clear that the edgel-data at corners is unreliable. Hence, as we do not go back to the image for further information, corners are localized by extrapolating the tangents at the ends of the corner segments. It is worthwhile to notice that tangent-discontinuity detection in the image-plane is equivalent to step-detection in the ψ–s plane. Similar to step-edge detection in the image, it is easier in the ψ–s plane to detect steps which are flat on their two sides than those which have large slopes, i.e. corners between straight lines are easier to detect than those between curves.

As the reader has probably noticed, we have been fairly sketchy in this section and the previous one. The reasons are two-fold. First, we were apprehensive about getting the reader bogged down in details without conveying to him the central ideas of the approach. Some of the issues we have dealt with, e.g. linking, conic-fitting, knot-placement, straight-line discovery, corner-detection, are involved enough to have spawned separate papers on each one of them. It is difficult to discuss all their nuances in the course of a single paper. Secondly, we hope to illustrate many of the concepts introduced in our discussion with an example in the next section.

IV. An Example

We now present a detailed example illustrating the working of our algorithms. Fig. 5-a is a (64 x 64) image of an industrial part. Fig. 5-b shows the edgels detected by the Nalwa Operator [Nalwa'84] mapped onto the connectivity-grid. Fig. 5-c is the resulting minimally-connected graph. We chose one of the linked edges to demonstrate our curve-fitting mechan-

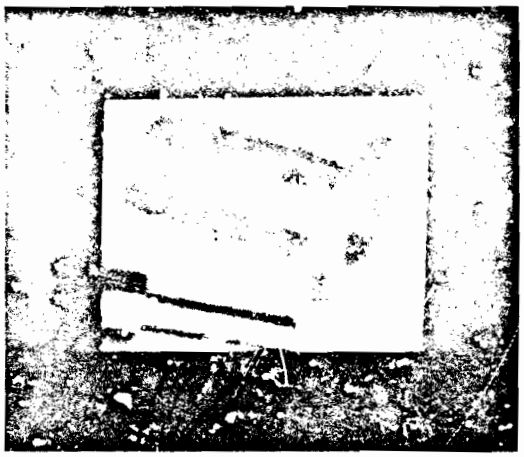


Fig. 5-a. Example : Original Image (64 x 64).

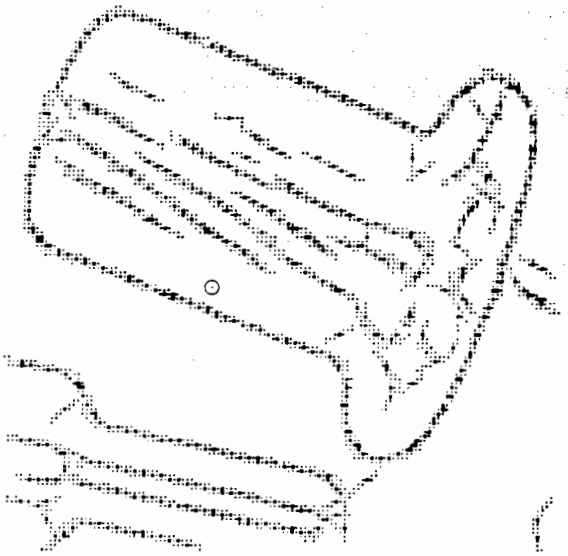


Fig. 5-b. Example : Detected edgels mapped onto the connectivity-grid (the largest dots indicate edgel-centers while the medium ones indicate edgel-extensions).

ism. Fig. 5-d shows the plot of the edgel-centers on the image-plane. Fig. 5-e shows the polygonal approximation to these edgels. Fig. 5-f is the resulting ψ–s plot. The detected straight-lines, inflection-points and tangent-discontinuities are marked on this figure. The initial choice of knots, obtained from a polygonal-approximation to the ψ–s plot, are also shown. Fig. 5-g is the image-plot corresponding to Fig. 5-f. Finally, in Fig. 5-h we show the fitted conic and the final set of knots. For reasons mentioned in the previous section, the localization of the corners is based on extrapolation of the tangents at the adjoining knots and not on the edgel-data there. The standard-

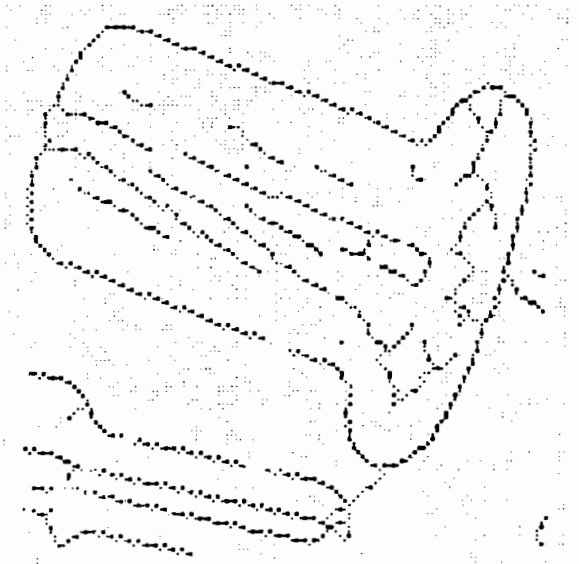


Fig. 5-c. Example : Minimally-connected graph obtained by thinning Fig. 5-b (the largest dots indicate edgel-centers while the medium ones indicate edgel-extensions).

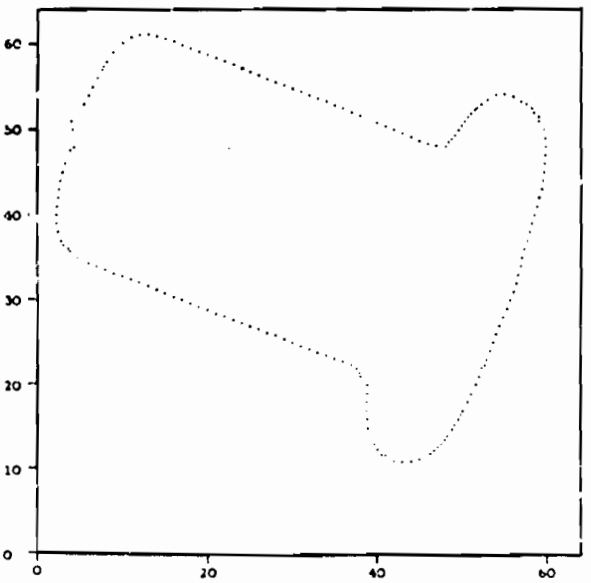


Fig. 5-d. Example : Edgel-centers of selected edge.

deviation of the error-distance of the edgel-centers from the fitted curve is less than 0.2 pixels and the maximum unsigned error is less than 0.5 pixels.

Fig. 6-a is an image of a blocks-world scene. Fig. 6-b is the corresponding mapping of the detected edgels onto the connectivity-grid and Fig 6-c shows the fitted curves.

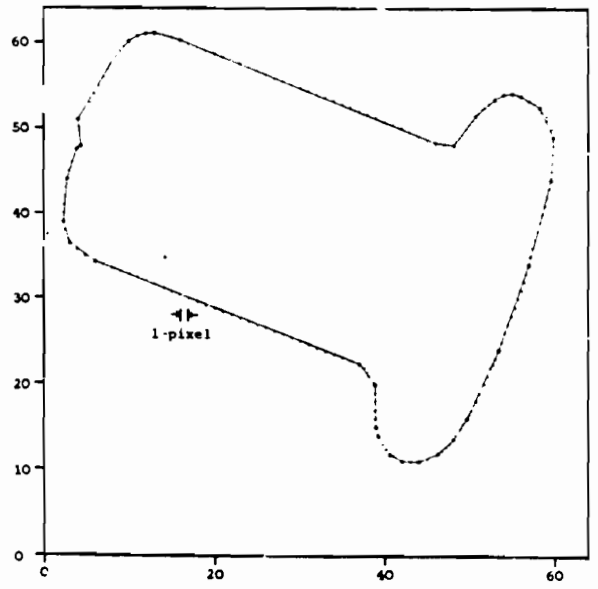


Fig. 5-e. Example : Polygonal-fit for data in Fig. 5-d.

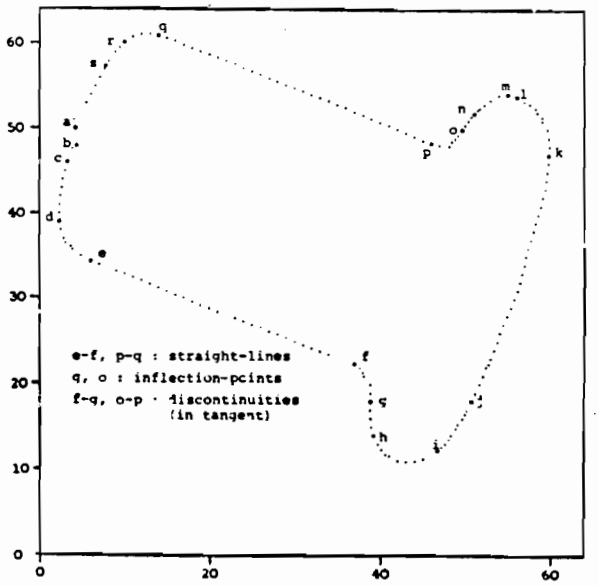


Fig. 5-g. Example : The image-plane, corresponding to Fig. 5-f, showing the initial placement of knots.

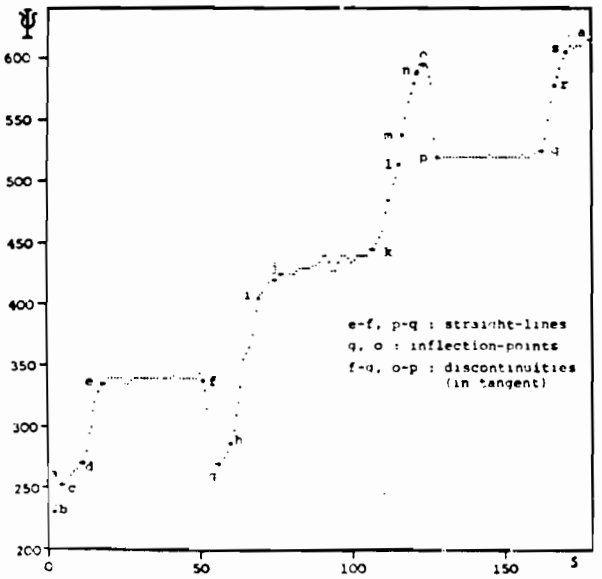


Fig. 5-f. Example : ψ - s curve with distance estimates obtained from the polygonal-fit in Fig. 5-e. The letters indicate the initial choice of knots based on a polygonal-fit in the ψ-s plane.

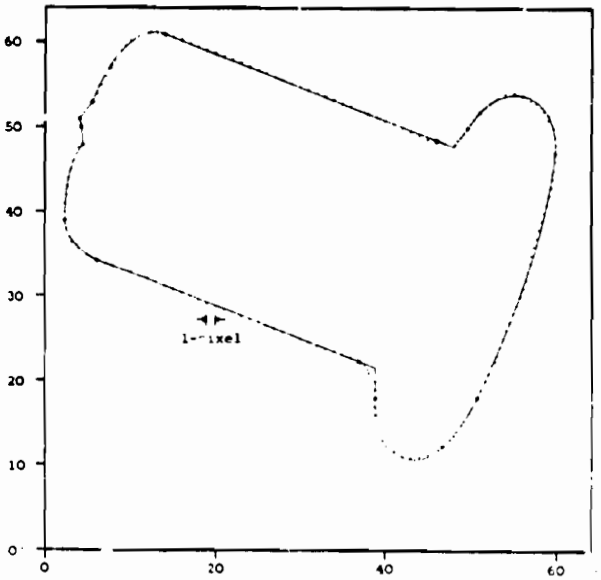


Fig. 5-h. Example : The final knot placement in the image-plane. The figure also shows the fitted straight-lines and guided-conics. Guided-conics are not fitted between knots corresponding to adjacent edgels because these conics are underconstrained. In the figure, such knots are connected by straight-lines.

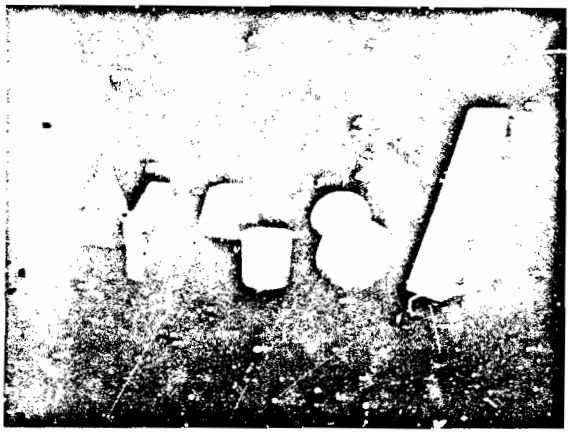


Fig. 6-a. Blocks-World Scene : Original Image (253 x 256).

Fig. 6-b. Blocks-World Scene : Detected edgels mapped onto the connectivity-grid.

Fig. 6-c. Blocks-World Scene : Linked edgels fitted with straight-lines and guided-conics.

V. Conclusion

In the course of this paper, we addressed some of the issues which arise when one attempts to aggregate edgels into extended edges and then describe these edges by a family of curves. Our edgel-linking algorithm is simple and has a local character. It relies only on edgel proximity and orientation. Unlike most previous attempts, there are few heuristics or thresholds. Conics splines were chosen for curve-fitting the aggregated edgel-data. Like any other set of curves, conics have their accompanying merits and failings. It is a non-trivial problem to estimate the distance of a point from a conic. We presented an approximate solution and indicated the condition for its validity. It was also shown why the exact solution is highly impractical. The error-criteria used for curve-fitting was the sum-of-squares of the distances of the edgel-centers from the conic. Our strategy to choose knots was justified on the basis of the Taylor-Series approximation to the ψ-s plot of the edgel-data. The final number and placement of the knots was based on the satisfaction of upper bounds on the mean-square-error and the maximum unsigned error for the edgel-data in the image-plane. Orientation compatibility between the edgel-data and the fitted conics was indirectly achieved. The other problems examined included, the discovery of straight lines and the detection of corners and inflection points. The algorithms were demonstrated with a detailed example.

It was not our intention to give detailed algorithms in this paper. Only a broad outline of our approach was presented. The details of the specific implementation are many and may vary.

We do not claim to have general-purpose and robust solutions to the linking and curve-fitting problems. Instead, we propose overall strategies which seem to offer promise. The problems considered here, in all their generality, are very involved. Some of the issues which demand further investigation include, the localization of edge-terminations, the detection and localization of junctions, the discovery and correction of systematic errors in the fitted curves and the development of mechanisms to correct for errors introduced due to the invalidity of the edge-detector hypotheses. We must also develop non-arbitrary mechanisms to choose the unavoidable thresholds. As the reliability of the detection and the localization of high-contrast edges is, in general, more than that for edges of low-contrast, it is advisable to vary the thresholds on the basis of the contrast. Consideration of the edgel-contrast may also be helpful during the linking stage.

Adequacy, and not efficiency, was our chief concern in the development of our algorithms. Hence, we have neither done a systematic analysis of the timings associated with the various components nor have we made an effort to improve them.

Appendix I : Approximate Distance of a Point from a Conic

Consider the conic,

$$C(x,y) = a.x^2 + b.xy + c.y^2 + d.x + e.y + f = 0$$

The distance, d_o, of a point, $P_o = (x_o, y_o)$, from the conic is measured along a perpendicular dropped from P_o onto $C(x,y) = 0$. We seek to estimate this distance.

The conic-section can be viewed as the intersection of the x-y plane with a surface $C(x,y)$ as shown in Fig. 7. The gradient at any point on the conic is perpendicular to the conic at that point. Hence, the distance, d_o, of a point, P_o, from the conic is measured along the gradient-direction at some point, P, on the conic. If P is close to P_o then it is reasonable to assume that the direction of the gradient at P_o closely approximates that at P. With this assumption it is straightforward to obtain a closed form solution for d.

The direction of the gradient at P_o is $\theta = \tan^{-1}\left|\frac{\partial C/\partial y}{\partial C/\partial x}\right|$, where θ is measured clockwise from the x-axis. Now consider the curve obtained from the intersection of $C(x,y)$ with a vertical plane passing through P_o and oriented in the direction of the gradient. Let us call its horizontal axis the d-axis and let the curve be represented as $Q(d)$. Further, let the origin be at P_o in this reference-frame. Fig. 6 illustrates a typical situation. Now let us write out the 1-D Taylor Series Expansion about the point P_o in terms of the directional derivatives of $C(x,y)$. We will use a subscript θ to indicate a directional derivative of $C(x,y)$ in the direction θ.

$$Q(d) = Q(0) + Q'(0).d + \frac{1}{2}.Q''(0).d^2 + \textit{higher order terms}$$

where

$$Q(0) = C(x_o, y_o)$$

$$Q'(0) = C'_\theta(x_o, y_o) = \pm\nabla C(x_o, y_o) = \pm\sqrt{\left|\frac{\partial C}{\partial x}\right|^2 + \left|\frac{\partial C}{\partial y}\right|^2} \quad \textit{at } (x_o, y_o)$$

$$Q''(0) = C''_\theta(x_o, y_o) = \{\cos(\theta)\ \sin(\theta)\} \begin{vmatrix} \frac{\partial^2 C}{\partial x^2} & \frac{\partial^2 C}{\partial x\,\partial y} \\ \frac{\partial^2 C}{\partial x\,\partial y} & \frac{\partial^2 C}{\partial y^2} \end{vmatrix} \begin{Bmatrix} \cos(\theta) \\ \sin(\theta) \end{Bmatrix} \quad \textit{at } (x_o, y_o)$$

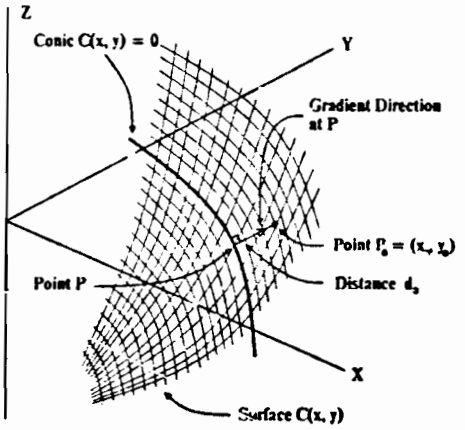


Fig. 7. The conic $C(x,y) = 0$ is shown as the intersection of the surface $C(x,y)$ with the x-y plane.

Substituting $\cos(\theta) = \frac{\frac{\partial C}{\partial x}}{|\nabla C|}$ *and* $\sin(\theta) = \frac{\frac{\partial C}{\partial y}}{|\nabla C|}$, *we get*

$$Q''(0) = -\frac{1}{|\nabla C|^2}\left\{\frac{\partial^2 C}{\partial x^2}\cdot\left|\frac{\partial C}{\partial x}\right|^2 + 2.\frac{\partial^2 C}{\partial x\,\partial y}\cdot\left|\frac{\partial C}{\partial x}\right|\cdot\left|\frac{\partial C}{\partial y}\right| + \frac{\partial^2 C}{\partial y^2}\cdot\left|\frac{\partial C}{\partial y}\right|^2\right\} \quad \textit{evaluated at } (x_o, y_o)$$

The higher order terms can immediately be seen to be 0 because $C(x,y)$ *contains only* 2^{nd} *order terms in* x and y.

Noting that $Q(d) = 0$ at the intersection of $Q(d)$ with the conic, it follows that the distance of P_o from the conic can be estimated from the roots of the quadratic $Q(d) = 0$.

$$d_o \approx \min\left\{\left|\frac{|\nabla C(x_o, y_o)|}{C''_\theta(x_o, y_o)} \pm \frac{\sqrt{|\nabla C(x_o, y_o)|^2 + 2.C(x_o, y_o).C''_\theta(x_o, y_o)}}{C''_\theta(x_o, y_o)}\right|\right\}$$

Note that the only approximation involved is that P_o is close enough to P so that the direction of the gradient at the two points is nearly the same. If the roots of the quadratic are imaginary, then this assumption is obviously violated. The line passing through P_o in the direction θ in the x-y plane, in general, intersects the conic at two points as shown in Fig. 8. We choose the smaller distance, invoking the

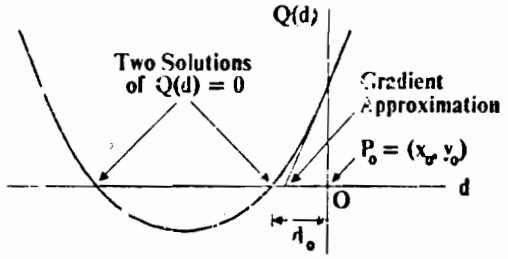


Fig. 3. A typical profile for the curve $Q(d)$ is shown. $Q(d)$ is obtained from the intersection of the surface $C(x,y)$ with a vertical plane passing through P_o and oriented in the direction of the gradient there.

closeness assumption once again. If the ratio $\left|\frac{C''(x_o,y_o)}{\nabla C(x_o,y_o)}\right|$ is small, then

$$d_o \approx \left|\frac{C(x_o,y_o)}{\nabla C(x_o,y_o)}\right|$$

This expression has been previously used [e.g. Turner'74] to estimate the distance. However, its limitations have not been noted. It is valid under the closeness assumption and the assumption that $\left|\frac{C''(x_o,y_o)}{\nabla C(x_o,y_o)}\right|$ is small. To see the error in using this expression, consider the conic $C(x,y) = (a.x + b.y + c)^2 = 0$, which represents a straight line. The distance estimated by the gradient approximation can easily be seen to be $\frac{|a.x_o + b.y_o + c|}{2\sqrt{a^2+b^2}}$. This, of course, is off by a factor of 2. The reader should observe that while our distance estimate is never less than the actual distance, the gradient estimate may be.

Appendix II : Exact Distance of a Point from a Conic

We indicate here, how one may proceed to find the exact distance of a point, $P_o = (x_o, y_o)$, from a conic, $C(x,y) = 0$. As discussed in Appendix I, this distance is measured along the gradient-direction at some point on the conic. Hence, we want to find a point, $P = (x,y)$, on the conic such that the vector connecting P and P_o is parallel to the gradient at P, i.e.

$$\frac{\partial C(x,y)}{\partial x}(y - y_o) - \frac{\partial C(x,y)}{\partial y}(x - x_o) = 0$$

As the partial derivatives of $C(x,y)$ are linear functions of x and y, we have a second order equation in x and y. Now if we express x and y in their parametric form [see Pavlidis'83],

$$x(t) = \frac{x_0 + x_1 t + x_2 t^2}{w_0 + w_1 t + w_2 t^2}, \quad y(t) = \frac{y_0 + y_1 t + y_2 t^2}{w_0 + w_1 t + w_2 t^2}$$

we get a quartic equation in the parameter, t. Quartic equations have closed form solutions, although they are too involved to be of practical use. Solving for t in the admissible range (typically 0 to 1), it is straightforward to find P and consequently the distance of P_o from the conic.

Acknowledgement

V.S.N. would like to thank his advisor, Tom Binford, for many useful discussions.

References

[Attneave'54] F.Attneave: "Some Informational Aspects of Visual Perception," Psychological Review, 1954, Vol.61, No.3, 183-193.

[Ballard & Brown'82] D.A.Ballard, C.M.Brown: "Computer Vision," Prentice-Hall Inc., Englewood Cliffs, 1982.

[Duda & Hart'73] R.O.Duda, P.E.Hart: "Pattern Classification and Scene Analysis," John Wiley & Sons Inc., New York, 1973.

[Nalwa'84] V.S.Nalwa: "On Detecting Edges," Proc: Image Understanding Workshop, New Orleans, Oct. 1984, 157-164.

[Pavlidis'83] T.Pavlidis: "Curve Fitting with Conic Splines," ACM Trans. on Graphics, Vol.2, No.1, Jan. 1983, 1-31.

[Pavlidis'82] T.Pavlidis: "Algorithms for Graphics and Image Processing," Computer Science Press Inc., Rockville, 1982.

[Turner'74] K.Turner: "Computer Perception of Curved Objects using a Television Camera," Ph.D. Thesis, A.I. Lab., Univ. of Edinburgh, Nov. 1974.

GC970145-24

Introducing a Smoothness Constraint in a Matching Approach for the Computation of Displacement Fields

P. Anandan and Richard Weiss
Computer and Information Sciences Departement
University of Massachusetts
Amherst, Ma 01002

Abstract

Correlation matching techniques for computing displacement fields from successive frames in a dynamic image sequence are known to be error-prone. Our previous work [Anan84] has been concerned with identifying the sources of these errors and computing a confidence measure. Here we formulate a smoothness constraint that is useful for improving unreliable matches in an image based on the reliable ones. We note the relationship between our formulation and the gradient based formulation of the smoothness constraint. We provide a hierarchical matching algorithm that includes our smoothness constraint and show preliminary results of applying our algorithm to real images.

1 Introduction

Intensity based techniques for the computation of displacements of image points between a pair of image frames rely on the similarity of the light intensity reflected from a scene location in the two frames. Such techniques include *gradient techniques* and *correlation techniques*.

In the traditional formulation of the gradient techniques, an intensity constancy assumption is used to provide a partial constraint on the displacement vector at each point in the image. A smoothness constraint on the displacement field is then used to uniquely determine the displacements [Horn81,Hild85] of all the points. The correlation techniques usually provide a unique displacement vector based on local matches, without incorporating any smoothness constraint. Hence, obvious local errors in the displacement fields computed by the correlation techniques go undetected.

In our previous work [Anan84], we identified some common sources of matching-errors during correlation and provided a scalar confidence measure with the estimated displacement at an image point. This measure, which indicated the reliability of the associated displacement estimate, used information already available during the correlation process and hence did not require significant additional computation.

This research was supported by DARPA under grant N00014-82-K-0464.

Although our previous measure was useful in determining the reliability of a displacement estimate, we indicated that since the displacement is a vector quantity, a vector valued confidence measure may be more appropriate. This paper provides such a measure and formulates a smoothness constraint on the displacement field that uses our new confidence measure.

Intuitively, we regard a displacement field to be "smooth" in an area of the image if its variation over the area is small. An example of a measure of the spatial variation of a displacement field is

$$E_{smooth}(U) = \int\int |\nabla U|^2 dx dy$$

where $U = (u, v)$ is the displacement vector at (x, y) (with u and v as its components in x and y directions respectively), and the domain of the integral is an area of the image (possibly the whole image). This is due to Horn and Schunck [Horn81], who use this measure in a gradient-based approach for the computation of optical flow. Another example of such a measure is provided by Terzopoulos [Ters84], which we will describe in detail later in this document.

The formulation provided here is the minimization of the sum of two errors, E_{smooth} and E_{approx}. Given a displacement field, E_{smooth} measures its spatial variation and E_{approx} measures its deviation from the initial displacement vectors computed by the matching process.

Our choice of the approximation error is based on the results of a matching process. Each displacement vector is represented in a convenient local ortho-normal basis (e_{max}, e_{min}), which are usually not parallel to (x, y). For a given displacement field U, the approximation error is a weighted sum of the deviations of the components (along these basis vectors) of the displacement vectors in the field from the corresponding components of known initial values (provided by the matching process). The weights are the components of the vector-valued confidence measures along the basis vectors.

Intuitively, the basis vectors e_{max} and e_{min} and the weights c_{max} and c_{min} can be understood as follows: At a point along an edge in the image, the basis vector e_{max} will be approximately oriented in the direction normal to the edge, and e_{min} will be oriented parallel to edge. At such

a point, the weight c_{max} will be large and the weight c_{min} will be small. On the other hand, in an area of the image with small intensity variations, both the weights will be small, whereas at a point along contour with high curvature, both the weights will be high. The precise definition of the basis vectors the weights are provided in section 5.

The precise form of the approximation error is

$$\begin{aligned} E_{approx}(U) &= \sum_{s,t} c_{max}(U \cdot e_{max} - D \cdot e_{max})^2 \\ &+ \sum_{s,t} c_{min}(U \cdot e_{min} - D \cdot e_{min})^2 \end{aligned}$$

where c_{max} and c_{min} are the weights, U is the displacement vector and D is the initial displacement vector provided by the correlation matching algorithm, at location (x, y) in the image. Here c_{max} and c_{min} indicate the confidence in the components of the displacement vector D in the directions e_{max} and e_{min} respectively.

In the rest of this paper, we develop this idea and formulate it as a relaxation process. We extend the analysis of Tersopoulos [Ters84] for the surface-reconstruction problem to our two dimensional minimisation problem. We follow a finite-element approach to solving the problem, primarily because this approach enables us to deal with arbitrary, known discontinuities in the displacement field.

We also incorporate the smoothness process within the framework of a hierarchical algorithm for the computation of displacement fields. Our hierarchical algorithm is similar to the multi-frequency coarse-fine techniques described in [Glas83,Anan84]. At each level of the hierarchy, we include the smoothness process after computing displacement estimates by correlation a matching process.

The relationship of this minimisation formulation to other formulations is described in section 2. In section 3, the conditions for the existence of a solution to this problem are explained, and in section 4 the finite element approach is used to provide a discrete relaxation algorithm. The source of the weights c_{max} and c_{min}, and the basis vectors e_{max} and e_{min} is discussed in section 5. The incorporation of this smoothness constraint in the hierarchical algorithm is described in Section 6. Finally, some experimental results are provided in section 7 and the scope for future work is discussed in section 8.

2 Relationship to other work

The minimisation problem posed in the previous section has its roots in other similar work in computer vision. In particular, the formulation of the intensity and smoothness constraints for the computation of optic flow fields by Horn and Schunck [Horn81], the related work by Nagel [Nage83,Nage84] and Hildreth [Hild83], and the surface reconstruction problem posed by Tersopoulos [Ters84] are closely related to our formulation.

First, we explain the relationship of our work to that of Horn and Schunck. Our smoothness constraint can be chosen to be the same as theirs. The intensity constraint they use is

$$\nabla I \cdot V + \frac{\partial I}{\partial t} = 0 \qquad (1)$$

where ∇I is the spatial gradient of the image intensity I at a point (x, y) in the image and V is the image-velocity at that point. Under the assumption that the time interval δt between successive image frames is small, we can approximate V by $D/\delta t$, where D is the displacement of an image-point, and $\partial I/\partial t$ by $\Delta I/\delta t$. Then, we can rewrite equation 1 as

$$\nabla I \cdot D + \Delta I = 0 \qquad (2)$$

Based on this, we can define an error

$$E_{approx} = \sum_{s,t} (\nabla I \cdot U + \Delta I)^2$$

which is zero when $U = D$ where D is any value that satisfies the equation 2 above.

This error can be rewritten as

$$E_{approx} = \sum_{s,t} |\nabla I|^2 (U \cdot e_{\nabla I} + \frac{\Delta I}{|\nabla I|})^2$$

where $e_{\nabla I}$ is the unit vector in the direction of ∇I.

From equation 2, we see that $D \cdot e_{\nabla I} = -\Delta I/|\nabla I|$. Considering our E_{approx} term described in section 1, if we set $c_{min} = 0$, $c_{max} = |\nabla I|^2$, $e_{max} = e_{\nabla I}$, our approximation error is the same as that of Horn and Schunck. Indeed, their intensity constraint simply provides values for one component of the displacement vector at any point in the image, viz., the component in the direction of the intensity gradient. This is because only the first order image intensity variations are considered.

Nagel [Nage83] points out that at certain areas in the image it may be possible to locally obtain values for both components of the displacement vector. This is usually true at points of high curvature along image-contours, or in textured areas, where the second order intensity variations are large and can be useful for obtaining the unique displacement vector. In section 5, we explain how a correlation matching algorithm behaves at such areas of the image.

Both our formulation and the formulation of Horn and Schunck can be regarded as a two dimensional version of the smoothness process presented by Tersopoulos [Ters84]. Tersopoulos is interested in visual surface reconstruction. This leads to the problem of smoothing a scalar variable u, the depth of the visible surface along each viewing direction. Approximate values for the depth and associated confidence measures are assumed to be known at some locations in the image. Similarly, approximate orientation of the surface and associated confidence measures are also assumed to be known at selected locations.

The problem is that of minimising the sum of three errors $E_{smoothness}$ and E_D, and E_O where $E_{smoothness}$ is due to a surface smoothness constraint, E_D is due to the known approximate depth values, and E_O is due to known approximate orientations of the surface. Tersopoulos considers two possible measures of the spatial variation of the

depth values. One is based on what is called a "membrane" model of the visual surface and the other is based on a "thin-plate" model of the surface. From our point of view, the two models are different simply with respect to the functional they minimise, which we will describe in detail in the next section. In attempting to extend this model to the two dimensional displacement fields, we ignore the orientation constraint, because we do not have any such information regarding our vector fields. For details, we refer the reader to [Ters84], Chapter 4.

In what follows, we provide a mathematical analysis of our problem and derive a method to iteratively solve it. Our mathematical analysis borrows heavily from that of Tersopoulos (see [Ters84] Chapters 5 and 6), with suitable modifications to deal with the two dimensional vector field.

3 Mathematical formulation

The problem of finding a smooth displacement field which approximates the estimated displacements on a discrete subset can be formulated as a minimization problem. That is, we are trying to find a vector field $U = (u, v)$ which minimises a quadratic functional $E(U)$ where $E(U) = E_{smooth}(U) + E_{approx}(U)$. As mentioned above, we consider two choices for E_{smooth}. E_{approx} measures how well U approximates the data given at a set of grid points (x, y) in D.

The two functionals which we consider have the property that under certain weak conditions there will always exist a unique minimum solution. The proof of this is based on the following theorem:

If $E(U) = B(U, U) - f(U)$, where B is a symmetric, bilinear form on a Banach space and f is a linear form, then E has a unique minimum if B is positive definite (in that case, E is said to be elliptic).

We will derive conditions on E_{approx} for each of the two smoothness constraints so that the positive definiteness condition is satisfied.

3.1 Smoothness constraint for thin plate model

This functional, which is minimised by an ideal thin plate, can be extended to a vector field $U = (u, v)$

$$\begin{aligned} E_{smooth}(U) &= \int\int (u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2) \\ &+ (v_{xx}^2 + 2v_{xy}^2 + v_{yy}^2) dx dy \end{aligned} \quad (3)$$

In order to apply the theorem, we decompose E into quadratic and linear terms. E_{smooth} is purely quadratic and E_{approx} has both linear and quadratic terms, so

$$B = B_{smooth} + B_{approx} \quad (4)$$

and

$$\begin{aligned} E_{approx}(U) &= \sum c_{max}(U \cdot e_{max} - D \cdot e_{max})^2 \\ &+ c_{min}(U \cdot e_{Min} - D \cdot e_{min})^2 \end{aligned} \quad (5)$$

So

$$B_{approx}(U) = \sum c_{max}(U \cdot e_{max})^2 + \sum c_{min}(U \cdot e_{min})^2 \quad (6)$$

We have confidence measurements c_{max} and c_{min} at each point, such that c_{max} is the confidence in direction e_{max} and c_{min} is the confidence in direction e_{min}, where e_{max} and e_{min} form an ortho-normal basis for the (x, y)-plane, and $0 \leq c_{min} \leq c_{max} \leq 1$.

B is positive definite if $B(U, U) = 0$ implies that $U = 0$. So we need to find conditions such that if $E_{smooth}(U) = 0$ and $B_{approx}(U) = 0$ then $U = 0$.

The smoothness constraint in particular implies that all of the second partial derivatives of u and v are 0, so u and v are linear

$$U = (ax + by + c, dx + ey + f) \quad (7)$$

We would like to show that all of these constants are zero. For example, if there are three non-collinear grid points which are corner points, then c_{max} and $c_{min} > 0$, and we have 6 independent equations for the coefficients. If on the other hand e_{max} always points in the same direction and c_{min} is always 0, then we cannot get independent equations, and we know that the answer will not be unique. Let $e_{max}(x, y) = (cos\theta(x, y), sin\theta(x, y))$ then $e_{min}(x, y) = (-sin\theta(x, y), cos\theta(x, y))$

The equations are of two types :

$$axcos\theta + bycos\theta + ccos\theta + dxsin\theta + eysin\theta + fsin\theta = 0 \quad (8)$$

or if $c_{min} > 0$

$$-axsin\theta - bysin\theta - csin\theta + dxcos\theta + eycos\theta + fcos\theta = 0 \quad (9)$$

The condition that will guarantee a unique solution is that one can find 6 independent equations from the set of equations defined by the points at which the vector field has been approximated. In particular, three non-collinear points with $c_{min} > 0$ or 6 points with $c_{max} > 0$ and which satisfy the following two conditions

1. e_{max} points in the same direction for no more than three points.
2. if e_{max} lies in the same direction for any subset of three points then they are not collinear.

3.2 Smoothness constraint for the membrane model

The smoothness constraint for this model is given by

$$E_{smooth}(u, v) = \int\int |\nabla u|^2 + |\nabla v|^2 dx dy \quad (10)$$

If $E_{smooth}(U) = 0$ then all of the first partial derivatives are 0, so U must be constant, i.e. $U = (a, b)$ Thus if two independent equations can be derived from B_{approx}, then U must be 0. This happens if there is a corner point or there are two points with different e_{max} vectors. Thus, this

condition implies that B is positive definite and therefore E has a unique minimum.

4 Solving the variational problem

In the previous section we showed that there is a unique smooth vector field which minimises the constraints; in this section we present methods to obtain a discrete approximation to this solution. The choice of domain and basis functions here is not one of the standard ones and is based on the approach of Terzopoulos. We have extended his results to two dimensional flow fields, and we present the masks which represent the solution. We intend to investigate a more standard choice of basis for the finite element technique.

We take as our discrete solutions piecewise polynomials; each polynomial being defined on a domain D in the plane. For the membrane model, we choose a triangular finite element as shown in figure 1. For the thin plate, we choose a domain which consists of a set of six points on a square grid (see figure 2).

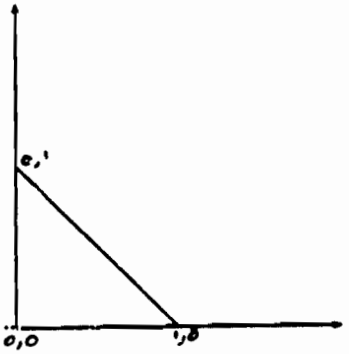


Figure 1: The finite-element domain for the membrane model

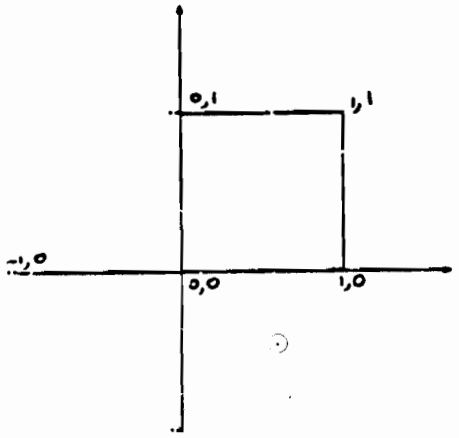


Figure 2: The finite-element domain for the plate model

There are two requirements for the finite element approach; that a unique solution exists for each particular grid, and that the solutions converge as the mesh size goes to zero. Our choice of the finite-elements follows that of Terzopoulos. He verifies that these elements satisfy these two requirements.

4.1 Computation of masks

In order to solve the discrete minimisation problem, linear equations in the values at the grid-locations are derived. These equations are used to update the values $U = (u, v)$ at a point in terms of its neighbors. This is commonly done by convolution with a mask; the masks for each of the two models are given below:

For the membrane model, the convolution mask is

$$\frac{1}{4} \times \begin{array}{ccc} & 1 & \\ 1 & 0 & 1 \\ & 1 & \end{array}$$

For the thin-plate model it is,

$$\frac{1}{20} \times \begin{array}{ccccc} & & -1 & & \\ & -2 & 8 & -2 & \\ -1 & 8 & 0 & 8 & -1 \\ & -2 & 8 & -2 & \\ & & -1 & & \end{array}$$

Let us denote the convolution mask as A. Also, let $\bar{U} = A * U$. Then, solving the discrete problem is the same as solving the following system of coupled equations:

$$\begin{aligned} (U - \bar{U}) &+ c_{max}(U \cdot e_{max} - \bar{D} \cdot e_{max})e_{max} \\ &+ c_{min}(U \cdot e_{min} - \bar{D} \cdot e_{min})e_{min} = \vec{0} \end{aligned} \quad (11)$$

for each point on the image grid.

4.3 Relaxation algorithm

There are a number of numerical-methods for solving the system of coupled linear equations described above. One of the simplest methods is the Gauss-Seidel relaxation algorithm. This is an iterative process, where during each iteration the value of U at each point in the image is solved in terms of the values of its neighbors.

In our case, we have

$$\begin{aligned} U^{n+1} = \bar{U}^n &+ \frac{c_{max}}{c_{max}+1}((D - \bar{U}^n) \cdot e_{max})e_{max} \\ &+ \frac{c_{min}}{c_{min}+1}((D - \bar{U}^n) \cdot e_{min})e_{min} \end{aligned} \quad (12)$$

where the superscripts denote the number of the iteration.

From a geometric point of view, this updating scheme can be regarded as choosing a point in the (u, v) space

that is a combination $\bar{U}^n$ and D. We illustrate this idea in figure 3. For convenience, we have chosen to represent the displacements in a cartesian coordinate system with its axes parallel to (e_{max}, e_{min}). The two key parameters are $c_{max}/(1 + c_{max})$ and $c_{min}/(1 + c_{min})$, which vary between 0 and 1, as c_{max} and c_{min} vary between 0 and ∞. Since $c_{max} \geq c_{min}$, the location of U^{n+1} will always be on or above the line joining D and $\bar{U}^n$ in figure 3. In particular, it can be seen that the location of U^{n+1} will be always within the triangle that is shown in that figure, moving towards the line joining D and $\bar{U}^n$ as c_{min} gets closer to c_{max}.

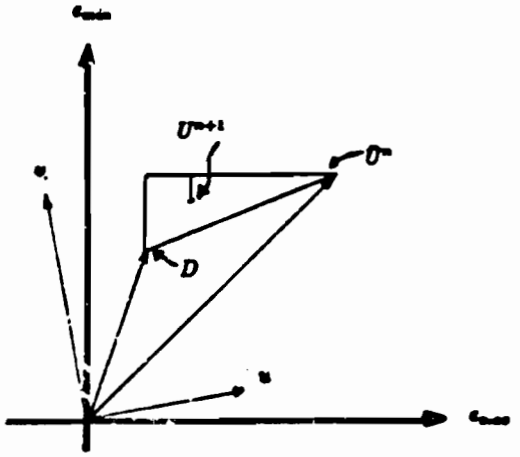


Figure 3: A geometrical illustration of the relaxation process

Finally, we note that although the Gauss-Seidel relaxation scheme is simple to implement, there are more efficient techniques to solve a sparse system of equations. Studying these and choosing an efficient, parallel scheme will be a part of our future work.

5 The basis vectors and the confidence measures

One of the key elements of our formulation is the approximation constraint. The proper choice of the ortho-normal basis vectors e_{max} and e_{min} and the confidence measures c_{max} and c_{min} are crucial to our model and to our algorithm. In this section we discuss how these measures can be obtained.

In attempting to choose these measures, it is useful to formalise our efforts in the form of a set of design considerations. These considerations are,

1. If both components of the two dimensional displacement vector are reliably known at an image location, both c_{max} and c_{min} should be large. If one component is reliably known and the other is not, then the c_{max} should be large and c_{min} small, and e_{max} should be oriented along the direction of the reliable component. If both are unreliable then both c_{max} and c_{min} should be low.

2. The computation of c_{max} and c_{min} should take minimum effort in addition to what is already necessary for the computation of the displacement vectors.

In our previous study on the behavior of *error* (sum of squared-differences) surfaces [Anan84], we noted that the shape of these surfaces contained useful information regarding the reliability of the displacement vectors. Below, we explain what we mean by these terms, and briefly summarise our major observations.

The process of matching by discrete correlation consists of choosing a window around a point of interest in one frame (the first window), and correlating the intensity values in that window with those in the candidate match windows in the second frame (the second window). The term *correlation* is used in a generic sense here. The measure that we prefer (for reasons explained in [Anan84]) is the sum of squared differences between corresponding pixel values (SSD).

By the term *error surface*, we mean the surface whose height is the SSD value corresponding to each possible displacement. Hence, for each location of interest in the first frame, we have available an error surface. The best match is, indeed, the displacement corresponding to the minimum height of the error surface (henceforth called the *pit*).

In our study of the error surfaces [Anan84], we also noted that the curvature of the error surface around the pit, taken along any direction seems to provide information regarding the reliability of the match along that direction. In particular, we observed the following :

1. At a corner point, the pit is sharp in all directions; correspondingly all the directional curvatures are high.

2. At a point along an edge, the error surface shows a long valley like structure, and the orientation of the valley corresponds to that of the edge. This means that the curvature in the direction parallel to the edge is low, whereas that in the direction perpendicular to the edge is high.

3. At a point in a homogeneous area all curvatures are low, and the error surface looks rather flat.

Figures 4, 5, and 6 show examples of error surfaces at a corner-point, an edge-point, and a homogeneous point respectively. The surfaces are shown inverted, in order to enhance visibility. Note that in a small area around the point with minimum error (the peak in these figures), the surfaces demonstrate the properties described above.

The directional curvatures of the error surface seem to satisfy the first of our design criteria. Further, the computation of the error surface is a necessary part of the

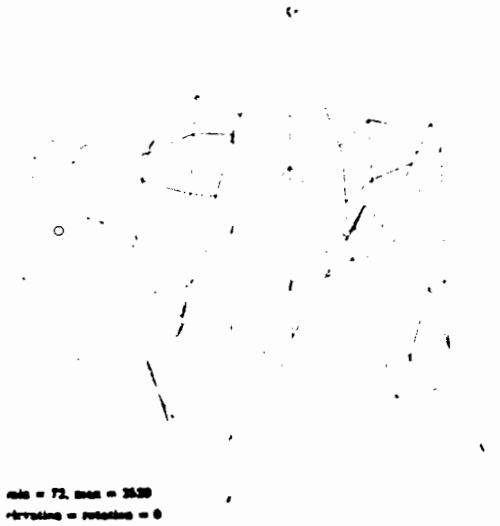


Figure 4: The error surface at a corner point

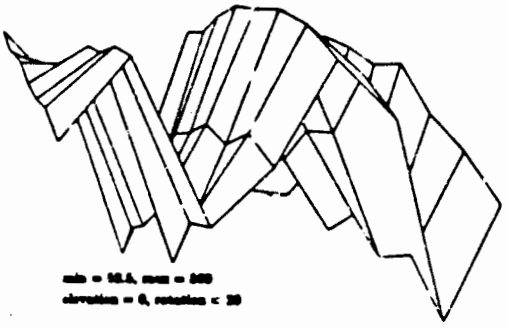


Figure 5: The error surface at an edge point

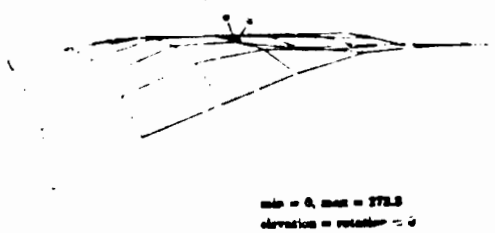


Figure 6: The error surface at a homogeneous point

matching process, and the computation of the curvatures requires only minimal additional effort. Hence, this meets both our design criteria. Based on these, our method of choosing c_{max} and c_{min} are as follows :

1. Compute the principal curvatures and the directions of the principal axes around the pit of the error surface. Let C_{max} and C_{min} be the principal curvatures and E_{max} and E_{min} be the unit vectors along the directions of the principal axes.

2. Compute the error corresponding to the location of best match. Let this be SSD_{min}.

3. We choose $c_{max} = E_{max}$ and $c_{min} = E_{min}$. For c_{max} and c_{min}, we have a number of choices:

   (a) We can simply take $c_{max} = C_{max}$, and $c_{min} = C_{min}$.

   (b) We can normalise the principal curvatures to lie within some desired range and to be scaled by a desired factor, and choose $c_{max} = NORM(C_{max})$ and $c_{min} = NORM(C_{min})$. Here $NORM(.)$ is the normalising function of the curvature.

   (c) We can normalise the principal curvatures using SSD_{min}. We can choose

   $$c_{max} = \frac{C_{max}}{k_1 C_{max} + k_2 SSD_{min} + k_3}$$

   and

   $$c_{min} = \frac{C_{min}}{k_1 C_{min} + k_2 SSD_{min} + k_3}$$

   This has the effect of making the confidence inversely proportional to the error corresponding to the best match. Usually this error is high when the local physical structure represented in the image changes, due to expansion, rotation, etc. of the image, or if the image has been corrupted by noise. Strictly speaking, correlation (or SSD) provides truly reliable matches only when these effects are absent. Hence, this type of normalisation takes into account problems that are unexpected, but influence the error measure.

The choice of the normalisation of the principal curvatures to obtain the confidence measures is a crucial factor. For the experiments described in this paper, we used a normalisation of the type discussed in 3(c) above, choosing $k_1 = 0, k_2 = 1$, and $k_3 = 100$. These choices were made on an empirical basis, although we observed that varying them did not significantly affect our results. This issue is open for further research.

6 A hierarchical matching algorithm with smoothing

The foregoing discussion regarding matching and smoothing and the associated confidence measures is based on the strong assumption that the displacements are small in magnitude compared to the scale of the image and the spatial rate of image intensity variations. It has been noted by other researchers [Burt83,Glas83] that in order to realistically deal with general images with larger displacements, it is desirable to use a hierarchical matching approach. This has the advantage of achieving both savings in computation and the use of image intensity variation at a scale appropriate for the amount of motion information. Our technique here is based on the hierarchical correlation algorithm described in [Glas83] and [Anan84].

Briefly, the hierarchical matching/smoothing algorithm is as follows:

- Each of the two images are processed with a set of band-pass filters and reduced in resolution. The band-pass filters are isotropic difference of Gaussian filters, one octave wide and one octave apart from each other. These are implemented based on the ideas suggested by Burt ([Burt81].
- The matching begins at a level of resolution corresponding to a displacement of less than 1 pixel. The process of correlation is described in [Glas83,Anan84]. Sum of squared-differences is used as the match measure. For the experiments of this paper, we chose square windows 5 pixels wide.
- After the matching at one level, confidence measures are computed as described in the previous section and smoothing is performed. For the experiments of this paper, we used a membrane model and finite element method for smoothing.
- The smoothed vector field is projected to the next finer-level image, based on the modified projection strategy described in [Anan84]. These are used as initial values for the matching/smoothing process at this finer level.
- This process is repeated at all levels up to the level of the image.

7 Experiments

We performed two experiments to illustrate our algorithm. These experiments are meant to be illustrative of the ideas presented in this paper. In this sense these results should be regarded as preliminary. However, it will be easy to see from these examples that the smoothness process reduces the errors in the initial match estimates.

The first experiment demonstrates the effect of applying our smoothness algorithm at a single level of resolution and computing a dense displacement field. The second experiment demonstrates the use of our confidence measure in a hierarchical matching/smoothing algorithm. In both cases, we used the membrane model for the smoothness constraint.

The first experiment was performed by using a single real image and generating a second image by digitally translating the first image. This enabled us to have "ground-truth" displacement data for comparison. The second experiment was performed using a pair of images from a *real* image sequence called *road-scene image sequence*.

7.1 Testing the single level smoothing algorithm

For this experiment, we used the image shown in figure 7 as the first of the pair of images. The second image was obtained by rotating the first image clockwise by 4 degrees on the plane of the image about the center of the image. The initial results of the matching process, prior to smoothing are shown in figure 8. The results after 100 iterations of the smoothness process are shown in figure 9. Although these results are useful to gain a qualitative understanding of the effect of the smoothness process, it is more useful to consider the statistics of the error vectors (i.e, the difference at each pixel between the true flow vector and the computed vector). These are shown in the table in figure 10.

Note that we have shown component-wise error statistics as well as the statistics of the length of the error vector. The component-wise statistics are based on a local decomposition of the error vectors in the directions of e_{max} and e_{min}. We use these in order to illustrate the point that most of the error is usually in the direction of e_{min}.

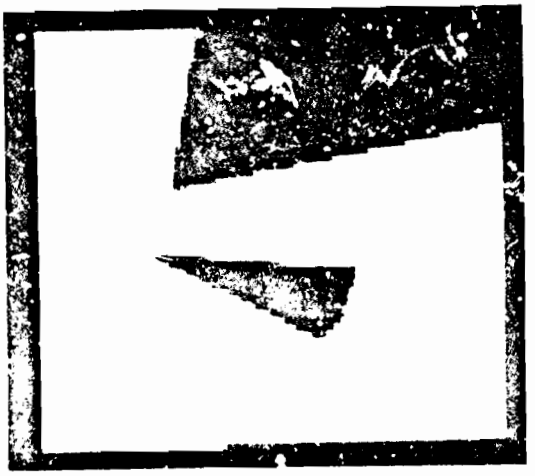


Figure 7: The input image for the single level experiment

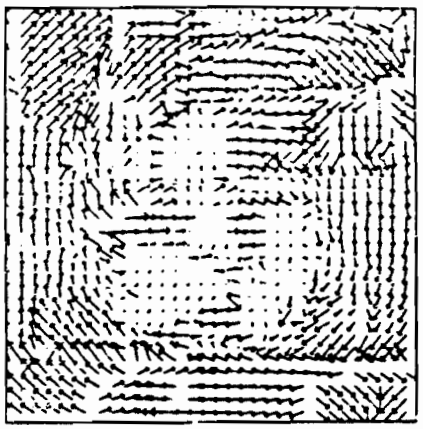


Figure 8: The initial displacement estimates provided by the matching process. Only a quarter of the displacement vectors are shown, in order to enhance visibility

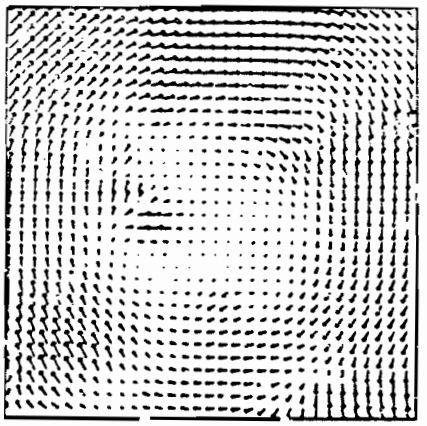


Figure 9: The displacement field after 100 iterations of smoothing using the membrane model

	err. along e_{max}		err. along e_{min}		err. mag	
	mean	std. dev	mean	std. dev	mean	std dev
initial disp.	0.0008	0.3410	0.0875	0.6875	0.2907	0.3977
final disp.	0.0041	0.1897	0.1106	0.3102	0.2961	0.2350

Figure 10: Table of error statistics for the single level experiment The table shows the statistics of the absolute values of the error component along e_{max} and e_{min} as well as those of the magnitude of the error vectors

7.2 Testing the hierarchical algorithm

Our second experiment involved performing the hierarchical matching algorithm with smoothing on a pair of images from the *road scene image sequence*, which is a sequence of images of a road scene obtained from a translating camera. The two images are shown in figures 11 and 12.

The displacement of any point between these two images is less than 10 pixels. This implies that four levels of

Figure 11: The first image of the road-scene image pair

Figure 12: The second image of the road-scene image pair

the hierarchy are sufficient for processing. At each level, we performed 10 iterations of the relaxation algorithm. This number was chosen arbitrarily.

In figures 13 and 14, we display the displacement fields at the four levels of the hierarchy. The dramatic improvement in the final results are the results of the integration of the matching and the smoothing process in the hierarchy. Smoothing at one level of resolution will not provide such improvement.

In figure 15 we display the confidence measures c_{max} and c_{min} as intensity images. The brightness of a point in these images are proportional to the confidence measures. We have also superimposed the unit vectors e_{min} on the figure containing c_{max}, in order to demonstrate that it is usually perpendicular to the edges in the image.

For the purposes of a closer scrutiny, in figure 16 we display the results only at the image-resolution.

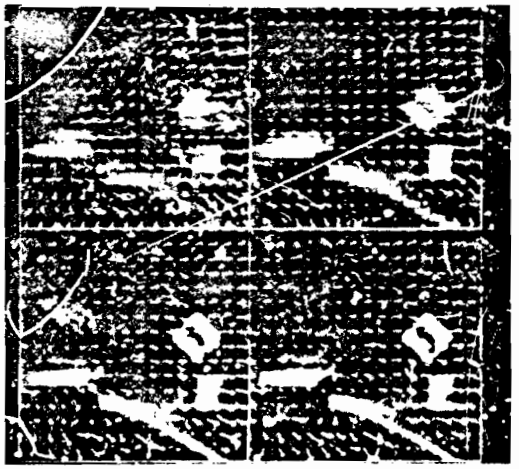


Figure 13: The results of the hierarchical matching algorithm without the smoothing process when applied to the road-scene image-pair.

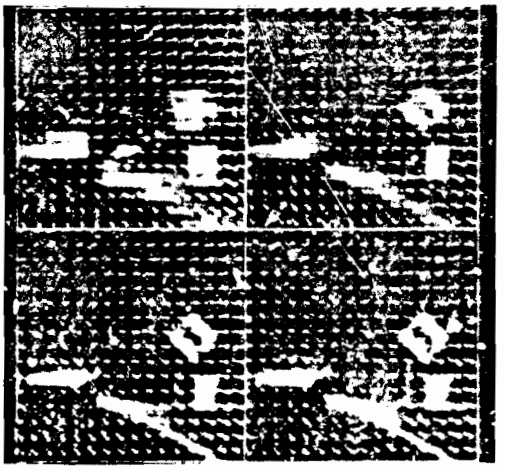


Figure 14: The results of the hierarchical matching/smoothing algorithm applied to the road-scene image-pair

In both the figures above, results at four levels of the hierarchy are shown. The background images are the low pass filtered versions of the first road image. The image resolution is 128 × 128 (shown in the bottom-right quadrant). At each level, except the 13 × 16 resolution in the top-left quadrant, we have shown only a sample of the displacement vectors. This was done so as to enhance visibility.

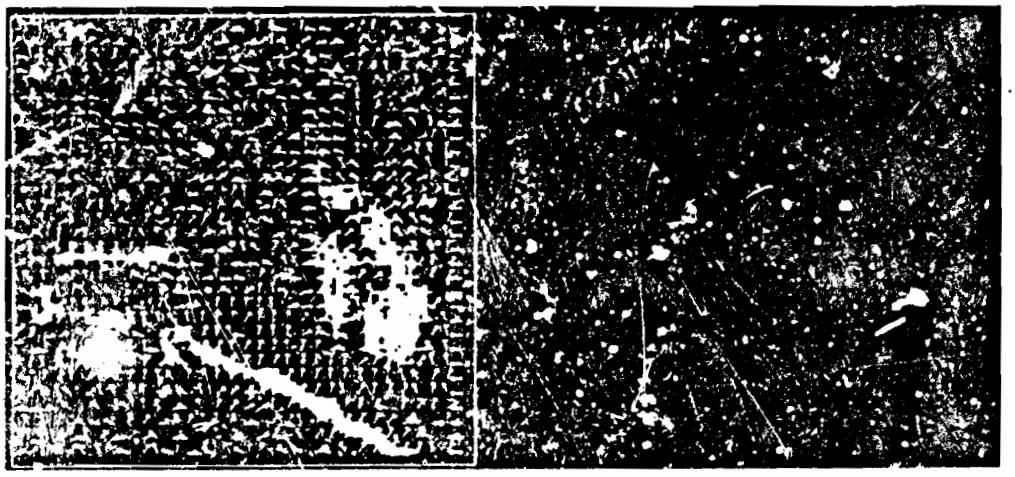


Figure 15: The confidence measures c_{max} (on the left) and c_{min} (on the right) at the finest-level of the hierarchy. These measures are displayed here as intensity images. The left figure also contains the unit-vectors e_{max}. Note that e_{max} vectors are perpendicular to the edges in the c_{max} image, which themselves usually correspond to the image edges.

Figure 16: The results at image-resolution for the road-scene image-pair

8 Future Work

In this paper, we provided an outline of a technique for incorporating a smoothness constraint in a matching algorithm. We also indicated how this may be incorporated in a hierarchical technique. Although we demonstrated these ideas with some experiments, many details of the algorithm are yet to be worked out. In particular, we feel the following areas need greater analysis and closer scrutiny.

1. The choice of the smoothness criterion. The membrane model as well as the thin plate model are heuristics for the type of variation we expect the displacement field to possess. It may be possible to limit the choice of the smoothness criterion based on an understanding of the geometric structure of continuous flow fields. Such analyses abound in the literature (e.g., [Adiv85,Waxm84]).

2. The normalisation of the confidence measures. One of the important factors that affect the outcome of the process is the validity of the confidence measure. At the outset, a conservative measure seems more appropriate, i.e., one that is more prone to doubt the reliability of the local match estimate. This will help us avoid the process from being misguided by incorrect values with high confidence. In general, a more thorough investigation is critical to make this process useful in real images.

3. One of the motivations behind choosing the finite element approach for solving the minimisation problem is that this approach allows us to deal with known motion and occlusion boundaries. Therefore, the recognition of such boundaries, and the inclusion of that information in the smoothness process is one of goals of future research. We seek an understanding of what information is available for recognising such events and how they may be utilised.

We believe that the hierarchical matching/smoothing algorithm is a useful technique for the computation of dense, reliable displacement fields in real images. Many issues remain to be solved. However, the approach takes into consideration the limits of its various components, and uses them appropriately. This provides us the motivation for further investigation along this line.

Acknowledgments

We wish to thank Dr. George Reynolds for his comments on the manuscript, and Frank Glaser for many valuable discussions. Thanks are also due to Brian Burns and Bill Guasso for their help in obtaining the figures for the paper.

References

[Adiv85] Adiv, Gilad, Determining 3-D Motion and Structure from Optical Flow Generated by Several Moving Objects, *IEEE PAMI*, Vol. PAMI-7, July 1985, pp. 384–401.

[Anan84] Anandan P., Computing Dense Displacement Fields with Confidence Measures in Scenes Containing Occlusion, *SPIE Intelligent Robots and Computer Vision Conference*, Vol. 521, pp 184-194, 1984, also *COINS Technical Report 84-32*, University of Massachusetts, December 1984.

[Burt81] Burt, P. J., Fast Filter Transforms for Image Processing, CGIP vol. 16, pp 20-51, 1981.

[Burt83] Burt, P. J., Yen C. and Xu X., Multi-Resolution Flow-Through Motion Analysis, *IEEE CVPR Conference Proceedings*, June 1983, pp. 246-252.

[Glas83] Glaser, F., Reynolds, G. and Anandan, P., Scene Matching by Hierarchical Correlation, *IEEE CVPR conference*, June 1983, pp. 432-441.

[Hild83] Hildreth, E. C., The Measurement of Visual Motion, *PhD dissertation*, Dept. of Electrical Engineering and Computer Science, MIT, Cambridge, Ma., 1983.

[Horn81] Horn, B. K. P. and Schunck B. G., Determining Optical Flow, *Artificial Inteiligence* Vol. 17, pp. 185-203

[Nage83] Nagel H. H., Displacement Vectors Derived from Second-Order Intensity Variations in Image Sequences, *Computer Vision, Graphics, and Image Processing*, 21, pp 85-117, 1983.

[Nage84] Nagel H. H., Constraints for the Estimation of Displacement Vector Fields from Image Sequences, *IJCAI-83*, Karlsruhe, W. Germany, pp 945-951, 1983.

[Ters84] Terzopoulos, D., Mutiresolution Computation of Visible-Surface Representations, *Phd Dissertation*, Massachusetts Institute of Technology, Jan. 1984.

[Waxm84] Waxman A. and Wohn K., Contour Evaluation, Neighborhood Deformation and Global Image Flow: Planar Surfaces in Motion, *CS-TR-1394* University of Maryland, April 1984.

GC9701460-25

On Surface Reconstruction Using Sparse Depth Data

by

Terrance E. Boult and John R. Kender†

Computer Science Department, Columbia University, NYC, NY 10027.

§0 Abstract

We report on our investigation into the problem of reconstructing a surface given a sparse set of depth samples. We discuss the formulation and recent history of the problem, its solution in abstract terms, identify some heretofor unmentioned assumptions in the formulation, and discuss possible alternative assumptions. We show that the space of algorithms solving the problem is large, and that there are least four major approaches, each of which is briefly discussed. Then we investigate implementation details of two of these approaches based on the use of reproducing kernel splines. We also discuss our future plans.

§1 Introduction

There are a number of applications in computer vision and robotics in which one desires to know the depth of an object at all points in some particular region, but the only information available is the depth (and possibly surface orientation) on a sparse set of points. To estimate the needed depth values we often attempt to reconstruct all or part of the object surface. Researchers have used a number of different approximations to the surface and an even greater number of representations for the reconstructed surface, e.g. see Allen (1985), Binford (1981), Mayhew (1982), Grimson (1981), Kender, Lee and Boult (1985), and Terzopoulos (1984). The last three of these formulate the problem in a very similar fashion, and it is this formulation that is discussed in this paper. This formulation defines the problem as attempting to reconstruct the surface that the human visual system would reconstruct if it were given the same sparse depth data (say, in a random dot stereogram). This formulates the problem as attempting to find the surface, from a class of surfaces, that minimizes a given unreasonableness function (usually a norm, or the sum of a norm and some other penalty function).

Marr (1977) noted that the problem of finding an algorithm to solve any problem can be broken up into four phases: formulation, solution in the abstract, analysis of alternative realizations of the solution, and the implementation and testing of some realization(s). Each of these phases is important, and each may affect the others, but they do provide a convenient way to break up the problem into smaller pieces. We shall use this division of a problem throughout this paper In Section 2 we give a precise formulation of the problems , briefly discuss its solution in the abstract. Then we question some assumptions implicit in the formulation (used in Grimson (1981), Terzopoulos (1984) and Kender, Lee and Boult (1985)), discuss alternative assumptions, and means for choosing between these assumptions.. In section 3, we address the third phase of problem solution by discussing four different realizations of the abstract solution. In section 4, we tackle the fourth and final phase of the division of a problem, by analyzing details of the methods of reproducing kernel splines, and seeing how their implementation details differ in various imaging situations. Section 5 addresses the future directions of our work.

§2 Formulation and Solution.

We shall examine two different reconstruction problems which have, unfortunately, been confused into one problem in the literature. We shall refer to them as the visual surface (hereafter VS) interpolation problem and the VS

† This work supported in part by: NSF grant MCS-782-3673, DARPA grant N00039-84-C-0165, an IBM fellowship, and an NSF Presidential Young Investigators award.

approximation problem. A naïve formulation of the VS interpolation problem would be to find "the best approximation" to a smooth surface, using only the knowledge of a number of given points, and requiring the surface to be interpolatory (i.e. passing through all the given data.) A similar formulation of the VS approximation problem would be to find "the best approximation" to a smooth surface using only the knowledge of a number of given points, but allowing the surface to be chosen freely otherwise. Because the problems are so similar, much of our discussion shall apply to both problems.

A major difficulty with these formulations is they are not well posed, inasmuch as the information does not uniquely determine the solution. In fact, given any set (of zero measure) of points on a surface, there are infinitely many surfaces interpolating those points (and of course, even more approximating them). To alleviate this problem we must somehow restrict the class of allowed surfaces, and give some method of ranking the "plausibility" of a surface.

A classical way of insuring that a problem has a unique solution is to use a functional on the surface as a measure of the "unreasonableness" of the surface, and restricting the class of allowed surfaces to make it a Hilbert or semi-Hilbert space with the unreasonableness functional as norm or semi-norm. This formulation insures that there exists a *unique* solution to the problem of finding a surface from the allowed class which minimizes the functional (and hence is the most reasonable). Throughout this paper we shall *assume* that this type of formulation is appropriate for our problems. Later in this section (see §2.3 and §2.4) we shall examine some issues associated with the choice of the Hilbert space and the unreasonableness norm.

§2.1 Formulation of the VS Interpolation Problem.

For the VS interpolation problem we choose to define "best approximation" in terms of minimal error. We *assume* that error can be measured by a norm with respect to the given class of functions. The norm might be the sup norm (i.e. the maximal difference between the actual surface and the approximation), or the L^2 norm (integral of the square of the difference at each point). The error may be measured in either a relative (e.g. error of 5%) or an absolute sense (e.g. the surfaces never differ by more than .1 mm) depending on the goals of the user. Finally the error may be measured in the worst case, or on the average (with respect to some measure).

Combining all of our assumptions, a precise formulation of the problem of the visual surface interpolation problem becomes :

> Let F_1, the space of allowed surfaces, be a Hilbert or semi-Hilbert space. Let F_2 be the elements of F_1 restricted to a finite domain Ω (since we are only interested in recovering a portion of a possibly infinite surface). Let $\Theta(f)$: $F_1 \rightarrow \Re$, be a functional measuring the "unreasonableness" of a surface (i.e. the more reasonable a surface f, the smaller $\Theta(f)$), where Θ is a norm or semi-norm on F_1. Now let $N(f) \equiv \{z_1,\ldots,z_n\} \equiv \{f(x_1,y_1),\ldots, f(x_k,y_k)\}$ be the given information (i.e the input is k depth values, see Kender, Lee and Boult (1985) for a discussion of allowed information.) Then the *visual surface interpolation problem* is to find $f^* \in F_1$, such that $\Theta(f^*) = \min_{g \in F_1} \Theta(g)$.

Kender, Lee and Boult (1985) show (as a special case of work on information based complexity (see Traub and Woz'niakowski (1980), or Traub, Wasilkowski and Woz'niakowski (1983)) that given the above formulation the surface minimizing the functional $\Theta(f)$ (i.e. the most reasonable surface) will also be the minimal error surface with respect to the class F_1 (i.e. the most accurate solution) for almost *any error* norm. This is what we refer to as the solution in the abstract. All we need to do is to actually calculate the surface of minimal norm itself.

§2.2 Formulating the VS Approximation Problem.

For the VS approximation problem we choose to define "best approximation" in terms both error and smoothness. We *assume* that error can measured by the sum, over all information points, of the distances between the approximating surface and the information points. For smoothness we shall assume that this is measured by a "unreasonableness" norm $\Theta(f)$; the smoother the surface the smaller $\Theta(f)$. This results in a formulation similar to the one for the VS interpolation problem. save there is a different functional to be minimized. The formal definition is:

Let F_1, the space of allowed surfaces, be a Hilbert or semi-Hilbert space. Let F_2 be the elements of F_1 restricted to a finite domain Ω (since we are only interested in recovering a portion of a possibly infinite surface). Let $\Theta(f)$: $F_1 \rightarrow \Re$, be a functional measuring the "unreasonableness" of a surface (i.e. the more reasonable a surface f, the smaller $\Theta(f)$), where Θ is a norm or semi-norm on F_1. Now let $N(f) \equiv \{z_1,\ldots,z_n\} \equiv \{f(x_1,y_1),\ldots, f(x_k,y_k)\}$ be the given information (i.e. the input is k depth values.) Then the *visual surface approximation problem* is to find $f^* \in F_1$, such that given $\mu > 0$, f^* minimizes $S(g)$ (with respect to all functions $g \in F_1$), where

$$S(g) \equiv \mu \cdot \Theta(g)^2 + \sum_{i=1}^{k} | z_i - g(x_i,y_i) |^2$$

Unfortunately this formulation does not currently allow direct application of the theories of information based complexity to make statements about the optimality (in terms of minimal error) of the algorithm. However, as with the VS interpolation problem, the VS approximation problem has been reduced to the problem of finding a surface from a class of surfaces that minimizes a functional.

Having given the general formulation of the problems, and their solution in the abstract (as functions minimizing $\Theta(f)$ and $S(f)$), it may seem that we have completed phases one and two of the solution of our problem. However, before we begin phase three, we should examine some of the assumptions, implicit and explicit, in the formulation of the problem.

§2.3 Discussion of Plausible Classes: How Smooth Should They Be?

In the definition of our problem we did not precisely define two important parameters, F_1 – the class of allowable surfaces, and $\Theta(f)$ – the functional measuring the "unreasonableness" of a surface. The two parameters are tightly coupled (recall that $\Theta(\cdot)$ is a norm or semi-norm over F_1) and so must be considered simultaneously.

Most of the interesting (class, norm) pairs $(F_1,\Theta(\cdot))$ give derivatives of the "surface", since we consider smoothness a part of reasonableness, and degree of differentiability can be considered a crude measure of smoothness. So far in the problem formulation, we have not explicitly assumed any amount of differentiability; we must make this part of the class definition. Note that if the surface is piecewise n times differentiable we can always assume the region Ω, which is part of the formulation, is such that the surface is smooth (i.e. it has differentiability of the assumed order) within Ω, and the reconstruction of the surface can be done piecewise. This however assumes that we know how to segment the surface – a difficult, and generally still open problem.

The question arises as to the number of derivatives that should be assumed. If the objects to be viewed are generated in a particular environment, such as a CAD/CAM manufacturing environment, we can often say with great reliability how many derivatives the surface should have. But in the unconstrained world we do not have such amenities. One way to approximate "reasonable" differentiability is to determine how many derivatives the human visual system can detect. Although it is unlikely the human visual system has an upper bound on the number of derivatives it can detect (e.g. it can tell if a surface has no more than 2 continuous derivatives), it may have a lower bound (e.g the surface has at least 3 continuous derivatives.) If we believe that the human visual system can detect discontinuities in n^{th} order derivatives, then our class should have at least n+1 (possibly many more) continuous derivatives, otherwise human observers might be able to detect discontinuities in the supposedly smooth surface. This does not rule out having explicit discontinuities, but rather unintentional ones due to the reconstruction algorithm.

To help answer the question of how many derivatives to assume, we have proposed a psychology experiment to measure the detectability of discontinuities in the derivatives of one dimensional curves. Preliminary results suggest that we need only consider discontinuities in the first second and third derivatives of a curve. Note that these results do not directly apply to the selection of a class of allowable surfaces. However, the number of variables in an experiment to measure the detectability of discontinuities in two dimensions make such an experiment unweildly. Some of the questions that must be answered before such a 2D experiment could be run include: What viewpoints do we assume? What lighting conditions do we assume? Do we texture the surface and if so how? Do we shade the surface?

If so, do we smoothly shade surfaces using a reflectance model based on surface normals or do we use one based on more derivatives of the surface? What method of display will allow us the needed accuracy in shading to correctly implement the reflectance model?

It seems intuitive to assume that the one dimensional result is a lower bound for the two dimensional problem: if you can detect discontinuities in the second derivative in a one dimensional curve, then you can detect them in a two dimensional surface as well. However, this intuition may be faulty because in the one dimensional experiment we have dense disparity data; but in two dimensions, due to sparsity of image edges relative to the scene, we may have only sparse depth data, although we may also have shading and texture data). Differences in the available information confound our intution. If depth or disparity data is most important for detection of discontinuities, then the 2D problem is more difficult: humans would not be able to detect the same discontinuities they could in 1D. If, on the otherhand, texture and shading aid in the detection of discontinuities, then the 2D problem may be easier. Other problems in the extension of the one dimensional result into two dimensions include: What is the analog of a 1D point discontinuity in 2D –a point or a line? If it is a line, then how does detectability of a point discontinuity in 2D relate to the detectability of discontinuities in 1D? Does the "direction" of the discontinuities matter, or is the detectability isotropic? To what extent do illumunation, shading and texture add or subtract from the ability to detect the discontinuity? Despite all the problems relating the one dimensional result to the two dimensional problem, they do directly apply when viewing the visible edge of a surface or a surface with ruled markings on it.

Although we do not have solid reasons for doing so, we shall follow our intuition and use the one dimensional result as a lower bound for the problem of choosing what is the appropriate two dimensional class for the VS interpolation or VS approximation problem.

§2.4 Discussion of Plausible Classes: Things to Consider and Examples.

Even with constraints from the 1D experiment mentioned above, we are still faced with an infinitude of (class, norm) pairs from which to choose. If the world is sufficiently restricted, as it may be in an industrial setting, then the choice of class may be easily made). In general, to make our choice we might appeal to physical analogies (e.g. the norm should measure physical bending energy of an ideal thin plate), computational biases (e.g. the space and norm should be computationally invarient and the mathematical definition of the class should be simple) or other adhoc assumptions (e.g. the (class, norm) should be such that the optimal surface exactly reconstructs low degree polynomials). However, most of these assumptions still leave infinitely many (class, norm) pairs satisfying them;they don't significantly cut down the choices.

How then do we choose? We propose to decide on the basis of a psychology experiment which subjectively ranks possible classes. Because we know the optimal error algorithm to recover surfaces in the (class, norm) pairs, any perceived difference between the various reconstructions *must* be due to differences in the model assumptions. Of course, we cannot subjectively rank a continuum of classes and norms, and so will use a finite subset of the infinite space of (class, norm) pairs covering a wide range of assumptions. It is quite possible that we shall find a number of the classes to be almost indistinguishable (or distinguishable, but none clearly a best fit). If this is the case then we can make our choice based on other considerations such as the computational complexity, some numerical properties, or some ad hoc assumption.

In the formulations of Grimson (1981), and Terzopoulos (1984), there was no assumption of a particular class of surfaces, just the assumption of a particular norm and the differentiability needed for this norm. In Kender, Lee and Boult (1985) they assume a particular class and norm. We now present a number of alternative classes and their associated norms (all of these will be used in our psychology experiments). For each class we briefly give some of the (ad hoc) reasons why one might choose this class. Each of these classes could be used in the formulation of our problems and each would offer different, at least mathematically, answers to those problems.

First we present notation that will aid in the discussion of the classes and norms. We shall let χ refer to an arbitrary class. We shall use the notation D_x^i to represent the differential operator $\partial^i(\cdot)/\partial x^i$, and D_y^j for $\partial^j(\cdot)/\partial y^j$. We drop i or j if they are equal to 1. In what follows the notation $D_x f(\alpha,\cdot)$ should be interpreted as

$$\left.\frac{\partial f}{\partial x}\right|_{(x,y)=(\alpha,y)}$$

(similar for derivatives with respect to y or for functions of one variable). Also $D_x D_y f(\alpha,\beta)$ should be interpreted as D_x applied to $D_y f(x,\beta)$ and evaluated at α. Finally we use the standart notation, $\prod_m$, to denote the space of bivariate polynomials with degree $\leq m$ in each variable.

A number of the classes we shall examine have a related semi-norm, the Sobolev semi-norm, $\Psi_m(\cdot)$, defined as

$$(2.1)\quad \Psi_m(f) = \left(\sum_{i+j=m} \iint_{\Re^2} \left| D_x^i D_y^j f \right|^2 \right)^{\frac{1}{2}}$$

(Surfaces minimizing this semi-norm are at times refered to as the m^{th} elastic medium.) For $m = 1$ this has the physical interpretation as the area of a membrane passing through the data; for m=2, it has the interpretation as the amount of bending energy in a thin elastic plate passing through known points. Hence minimizing $\Psi_2(f)$ is equivalent to finding a function which passes through the data points and has minimum bending energy. The reader familiar with the works of Grimson (1981) and Terzopoulos (1984) will recognize this as the same functional that their relaxation algorithm attempts to minimize. Grimson argues that this functional is minimized by the human visual system, though he does not present any psychophysical basis for his conclusions. Note his arguments only consider isotropic second order operators.

Before we begin giving examples,we would like to point out that there are other classes not presented here that are plausable classes for the problem, and shall be considered part of the psychology experiment. Out intention is to indicate that even on purely mathimatical grounds, many classes for interpolating or approximating surfaces exist. The reader interested in more classes should see Boult (1985c) or Boult (1986) for more details, including the reproducing kernels for the four faimlies of classes we shall now consider.

§ 2.4.1 A simple family of classes using an isotropic, physically meaningful semi-norm defined over $\Re^2$.

Duchon (1976) called this infinite family of classes $D^{-m}L^2$, and it is defined as the space of functions which have all partial derivatives of order m in $L^2(\Re^2)$. Then given that $m \geq 2$ we have that $D^{-m}L^2/\prod_{m-1}$ is a Hilbert space with the m^{th} Sobelev semi-norm as a norm. This class is defined to be as general as possible while allowing the use of the m^{th} Sobelev semi-norm, which a physical interpretation. This class is large, and therefore the optimal algorithm will have larger error than in some of the following, smaller classes. Furthermore, because the class is defined over $\Re^2$ we cannot use prior knowledge about the boundry of the surfaces in the class to reduce our error.

For the choice m=2, this class is implicitly considered in Grimson(1981) and Terzouplos (1984). It has their choice of an unreasonableness functional, and requires the minimal assumptions necessary to use that functional.

§ 2.4.2 A family of classes using an isotropic, physically meaningful semi-norm defined over $\Re^2$ with a somewhat band limited function space.

This family of class was first used and discussed in Duchon (1976). This family is defined with 2 infinite parameters (one of them continuous) and has the property that we also assume that the Fourier transform of all functions in the space is limited. (Given the discrete nature of the computer vision domain, band limiting the function space seems appropriate; we do not want surfaces oscillating wildly in between our discrete samples).

To be precise we define $D^{-m}H^s$, for $s \leq 1$ (s real), to be the space of functions which have all partial derivatives of order m in H^s, where H^s is the Hilbert space of functions with tempered distributions ν such that the Fourier transform, $\underline{\nu}$, of ν satisfies

$$\iint_{\Re^2} |\tau|^{2s} \left| \underline{\nu}(\tau) \right|^2 d\tau < \infty$$

Then Duchon shows that $D^{-m}H^s$ is a semi-Hilbert space with $\Psi_m(\cdot)$ as the associated semi-norm. These classes are only slightly smaller (depending on s) than the classes D^{-m}

L^2, mentioned above. Again because the class is defined over $\Re^2$ we cannot use prior knowledge about the boundry of the surfaces in the class to reduce our error.

§ 2.4.3 A class using an isotropic, physically meaningful semi-norm defined on a finite disk.

This class of functions uses the second Sobelev semi-norm defined by (2.1) for $m = 2$ and has as its domain finite disk. The advantage of being defined on a finite disk is that it results in a smaller class, thus a possibly smaller error

Let Ω be a disk with radius r centered at the origin and let H be the set of twice differentiable continuous functions $f: \Omega \to \Re^2$, such that the first and second order partial derivatives are elements of $L^2(\Omega)$.Define:

$$\chi = \left\{ f \in H: \oint_{\partial\Omega} f(x,y) = \oint_{\partial\Omega} x \cdot f(x,y) = \oint_{\partial\Omega} y \cdot f(x,y) = 0 \right\}$$

Then Atteia (1966) shows that χ is a semi-Hilbert space with $\Psi_2(f)$ a semi-norm with nuil space $\{1,x,y\}$ (i.e. χ / Π_1 is a true Hilbert space with $\Psi_2(\cdot)$ as a norm). One can interpret the definition of the class as limiting the space to functions that are "balanced" with respect to the boundrary. In fact, if the graph of a function on boundary of the disk were considered to be a thin wire, this class consists of those functions whose center of gravity for the wire is exactly the orgin.

§ 2.4.3 A class using an non-isotropic semi-norm, defined on an arbitrary rectangle in $\Re^2$, and a norm incorporating prior knowledge on a finite set of points.

This is one of a number of families that are not necessarily isotropic (though for brevity, we do not consider the others.). If there is some physiological or psychological reason (e.g. the horizontal positioning of the eyes) that results in human perception of smoothness being non-isotropic, then we might use one of these classes. Futhermore the norm in these classes can be tailored to reflect explicit knowledge about how the class of surfaces behaves on a finite set of points. Let $R^{m,n}$ be the Hilbert space containing functions defined on $\Omega = [a,b]\times[c,d]$ such that for all $f \in R^{m,n}$ we have :

$D_x^i f$ is continuous for $i = 1, \ldots, m-1$;

$D_y^j f$ is continuous for $j = 1, \ldots, n-1$;

$D_x^{m-1} f$ and $D_y^{n-1} f(x,y)$ are absolutely continuous; and finally

$D_x^m f$ and $D_y^n f$ are in $L^2(\Omega)$.

Let $\{x_1, \ldots, x_m\}$ and $\{y_1, \ldots, y_n\}$ be distinct points in [a,b] and [c,d] respectively. Then Arthur (1974), shows that $(f,f)^{1/2}$ is a semi-norm on $R^{m,n}$, where the inner product (f,g) on $R^{m,n}$ is given by:

$$(f,g) = \sum_{i=1}^{m} \sum_{j=1}^{n} f(x_i,y_j) \cdot g(x_i,y_j)$$

$$+ \sum_{j=1}^{n} \int_a^b D_x^m f(x,y_j) \cdot D_x^m g(x,y_j)\, dx$$

$$+ \sum_{i=1}^{m} \int_c^d D_y^n f(x_i,y) \cdot D_y^n g(x_i,y)\, dy$$

$$+ \int_a^b \int_c^d D_x^m D_y^n f(x,y) \cdot D_x^m D_y^n g(x,y)\, dy\, dx$$

The null space associated with this semi-norm is

$$\left\{ f \in R^{m,n} \text{ such that } f(x,y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \beta_{i,j} \cdot x_i \cdot y_j; \quad \beta_{i,j} \text{ constant.} \right\}$$

which is of finite dimension m·n.

These classes are but a few of the many that exist. Others have yet other properties (e.g. the norm is such that the result of minimizing it is exactly reconstructs low degree polynomials; regions of different smoothness can be incorporated into the class definition, more prior knowledge can go into the definition of the norm, etc.) Therefore, before the first and second phase of algorithm development is complete, a rationale for the choice of the (class, norm) pair must be found. It should be clear that the choices are many, and even an appeal to human perception is ultimately a statement of preference

§3 Analysis of Alternative Realizations

The third phase in the solution of a problem is the analysis of alternative realizations of the solution. There are a number of ways that one can attempt to find a surface from a Hilbert space that minimizes a functional. Of these we shall examine four different methods and discuss their advantages and disadvantages. These methods and the subsections in which they are discussed are:

§3.1 Discretization of the problem using variational principles and then discrete minimization using classical minimizaion techniques, as in Grimson (1981);

§3.2 Discretization of a partial differential equation for - mulation of the problem, again using a variational approach, and then use of discrete finite element approximation solved with a multigrid approach, as in Terzopoulos (1984);

§3.3 Direct calculation using reproducing kernel splines for the Hilbert space including the null space of the norm, as done by Duchon (1976). (We shall refer to these as semi-reproducing kernel splines);

§3.4 The separation of the Hilbert space and the null space of the norm with direct calculation using the reproducing kernel of the resulting quotient space, as was done by Meinguet (1979). (We shall refer to these as quotient reproducing kernel splines).

It is beyond the scope of this investigation to discuss all the details of each of these realizations; we shall only discuss the relative advantages and disadvantages. For a more thorough comparison of Grimson (1981) and the quotient reproducing kernel spline approach, see Boult (1985b).

§3.1 Discretization and classical minimization.

Grimson (1981) proposed realizations that solve discrete versions of the VS problems (one method each for the VS interpolation and the VS approximation problems) by employing variants on classical minimization techniques. Some advantages of his methods are:

+ The minimization techniques employed are reasonably well studied (e.g. they are known to be numerically stable and there are upper bounds on the rate of convergence; etc...);
+ It employs "computational modules" that are relatively simple;
+ It may be biologically feasible and may even model the human visual system;
+ Almost all calculations can be done in a purely local manner.

However, the methods also have the following disadvantages:

- The rate of convergence for the iterative relaxation used in minimization is slow;
- A good stopping criterion for the iteration is lacking;
- The use of a grid representation limits the applicability of the method, and also makes it dependent on the scale, translation, and rotation of the data;
- There is difficulty in deriving the computational modules if the boundary of the object is irregular;
- There is difficulty in adapting realization to other unreasonableness norms;
- There is "slower convergence" away from information points and near the grid boundary;
- Because it is a discrete version of a continuous problem, there may not exist a unique solution to the discrete problem.
- The minimization does not take into consideration a particular space of smooth functions hence, it may arrive at a solution that is not even from the appropriate functional class.
- If one changes the measure of reasonableness, the computational modules must be rederived and the numerical properties of the minimization must be reestablished.

§3.2 Discretization and multi-grid minimization.

Terzopoulos (1984) proposed realizations that also solve discrete versions of the problems. For a particular reasonableness measure, he relates the problem to a physical problem (thin plate interpolation) and uses variational principles and finite element-like methods to define an algorithm to calculate the surface of minimal energy (minimal norm). His methods employ a multi-level grid approach to the minimization stage of the problem, greatly increasing the computational efficiency of the algorithm. The advantages of Terzopoulos's realizations are:

+ The methods are far more computationally efficient then Grimson's methods.

+ The methods have the ability to measure error differently at each data point;
+ They have the ability to deal with discontinuities (given apriori) in the surface and its orientation,
+ They use a pyramid representation of the surface, which is convientent in some vision applications;
+ All the computation on each level of the pyramid is local;
+ There is only simple communication between levels (with a two directional flow).

Some of the disadvantages of Terzouplos' methods are:

- A good stopping criterion for the iteration is lacking;
- The use of a grid representation limits the applicability of the method, and also makes it dependent on the scale, translation, and rotation of the data.;
- There is difficulty in adapting realization to other unreasonableness norms;
- It may be difficulty to derive the computational modules if the boundary of the object or the known discountities is irregular;
- There is "slower convergence" away from information points and near the grid boundary;
- The numerical stability and convergence rates for the multi-grid approach are not apparent;
- The methods solve a discrete version of the proble suffering from the same problems as Grimsons';

§3.3 Minimization with Semi-Reproducing Kernel Splines

The realization using semi-reproducing kernel splines uses the mathematical properties of the kernels to exactly solve the continuous problems we formulated in section 2. Semi-reproducing kernel splines are defined in terms of the reproducing kernels for the semi-Hilbert space, see Duchon (1976), Boult (1985a, 1986), or Kender, Lee, and Boult (1985). The major computational component of the method is the solution of a dense linear system of equations (see section 4). Advantages of this approach include:

+ The solution of the linear systems is a well understood topic;
+ The algorithm results in a functional form for the surface allowing symbolic calculations (e.g. differentiation or integration);
+ The method is independent of the shape of the bound - ary of the object;
+ There is *no* problem with "slower convergence" away from information points, or near the object boundary;
+ The algorithm can efficiently allow updating the information (e.g. adding new data points, removing a point or changing a previous information value);
+ No iteration is needed - the amount of computation is fixed and depends only on the position and number (not value) of the information;
+ The kernels are independent of the information;
+ The method is easily adapted to other spaces of functions.
+ Because it solves the continuous problem, it is guaranteed a unique solution.
+ If the norm is isotropic, then the kernel is rotationally, translationally and scale invarient.
+ For sparse data this realization is more efficient that Grimson's or Terzopoulos's approach.
+ The linear system is symmetric; if the data falls on a regular grid the matrix is block toeplitz, which admits particularly efficient solutions;
+ The kernels are simple, relative to the kernels of the true reproducing kernel representation. (This is not apparent from our discussion, see Boult (1985a) for more details);

The disadvantages are:

- The resulting linear system is dense and indefinite which limits the approach we can use to solve it (in fact the system will always have d negative eigenvalues, where d is the cardinality of the nullspace);
- Although reproducing kernels exist for all Hilbert spaces, deriving them may be difficult. (However they are known for a large number of interesting classes, including all classes presented in section 2);
- This method may not be biologically feasable, due to the implicit global communication demands..

§3.4 Minimization with Quotient Reproducing Kernel Splines

A fourth realization of the optimal algorithm uses quotient reproducing kernel splines. The splines are defined in terms of the reproducing kernels for the quotient space –the original space with the null space of the norm removed. Thisis very similar to the realization using kernels for the whole space, except that the kernels are more complicated, (see Meinguet (1979a, 1979b) or Boult (1985a, 1985b, 1986) for more details). With respect to the other

realizations, this approach has all the advantages and disadvantages of the semi-reporducing kernel splines. The main advantage (over the semi-reporducing kernel splines) is:

+ The resulting linear system is positive definite. This is a important property from the numerical analysis point of view, insuring the numerical stability of algorithms for the solution of the system, and increasing the number of algorithms that can be used to solve it.

The disadvantages are:

- The condition number of the system appears to be significantly larger than that for the semi-reproducing kernel splines;
- The kernels themselves are much more complex, as compared to the kernel functions for the semi-reproducing kernel representation. Therefore the time required for calculation of the surface at each point is greater.
- The methods must explicitly calculate a unisolvent set of data, and functions interpolating the unisolvent set of data; in general, these can not be precomputed (see §4.2).

§4 Using Reproducing Kernels.

Now we are ready to explore the fourth phase of problem solution: the implenentation details. In what follows we first examine two separate reproducing kernel based methods (described in §3.3 and §3.4 above), giving the necessary linear systems in sections §4.1 and §4.2. In §4.3 we examine six different imageing situations briefly, describing which of the two method and which implenentation specific details are most suitable to the situation.

For reproducing kernel based methods to be applicable it is sufficient that F_1 (defined in section 2) be semi-Hilbert and $\Theta(f)$ the associated semi-norm with null space Π_m. To insure uniqueness of the solution we must assume that N_k contains a Π_m unisolvent subset (i.e. there exists a set J (a subset of 1...k) of indices, and a set of information points $\{x_j,y_j\}_{j \in J}$ and associated information values z_j such that there exists a unique $p_j(x,y) \in \Pi_m$ satisfying the condition $p_j(x_j,y_j) = z_j$ for all j in J. (For example, for the case presented in Kende, Lee, and Boult (1985) (see also §2.4.2), this amounts to having four non-coplanar points.)

§4.1 Semi-keproducing Kernel Splines.

Duchon, extending the work of Atteia (1976) to the case of semi-Hilbert spaces, noted that the solution to the problem of finding an interpolating function (or an approximating function as defined in section 2) of minimal norm in the semi-Hilbert setting could be written down in terms of $K((x,y);(s,t))$, the reproducing kernel of F_1. We give the derivation in general terms.

Given a reproducing kernel $K((x,y);(s,t))$ for F_1 we can write the interpolatory spline that minimizes $\Theta(f)$ as

$$\sigma_1(x,y) = \sum_{i=1}^{k} \alpha_i \cdot K((x,y);(x_i,y_i)) + \sum_{i=1}^{d} \beta_i q_i(x,y) \qquad (4.1)$$

where $\{q_i\}_1^d$ (d = cardinality of the set J) is a basis for Π_{m-1}, the null space of $\Theta(f)$. The coefficients $\{\alpha_i\}$ and $\{\beta_i\}$ of the interpolating spline can be determined from the solution of a (k+d) by (k+d) dense linear system.

Recalling the definition of $N(f) = \{z_1, \ldots z_k\} = \{f(x_1,y_1),\ldots, f(x_k,y_k)\}$, where (x_i,y_i) are the location of the function (depth) values we can express this linear system as follows:

$$\sum_{i=1}^{k} \alpha_i \cdot K((x_j,y_j);(x_i,y_i)) + \sum_{i=1}^{d} \beta_i q_i(x_j,y_j) = z_j, \quad j = 1,\ldots,k \qquad (4.2)$$

and

$$\sum_{i=1}^{k} \alpha_i q_j(x_i,y_i) = 0, \quad j = 1,\ldots,d.$$

Duchon (1976) shows that this representation yields a σ_1 that minimizes the functional $\Theta(\sigma)$ and is unique if the set $z_i = f(x_i,y_i)$, $i = 1,\ldots,k$ contains a Π_{m-1} unisolvent subset. This then solves the VS interpolation problem.

If we wish to solve the VS approximation problem we have the same representation (4.1) for the spline; however, the coefficients $\{\alpha_i\}$ and $\{\beta_i\}$ of the approximating (smoothing) spline can be determined from the solution of a (k+d) by (k+d) dense linear system given by:

$$\sum_{i=1}^{k} \alpha_i \cdot K((x_j,y_j);(x_i,y_i)) + C_K \mu \alpha_j + \sum_{i=1}^{d} \beta_i q_i(x_j,y_j) = z_j, \quad j = 1,\ldots,k \tag{4.3}$$

and

$$\sum_{i=1}^{k} \alpha_i q_j(x_i,y_i) = 0, \quad j = 1,\ldots,d.$$

where q_i is as before, C_K is a constant dependent on the kernel and μ is the parameter used in the definition of the approximation problem. For a more complete treatment, including examples of systems for particular kernels, see Boult (1985a) or Boult (1986).

§4.2 Quotient Reproducing Kernel Splines.

Another realization of minimizing a functional over a Hilbert space is due to Meinguet. We use the fact that we can separate the space F_1 into $X_0 \oplus \Pi_m$, where Π_m is the null space of $\Theta(\cdot)$, $X_0 = \{g \in F_1 : g(x_j,y_j) = 0, \forall j \in J\}$ and $\oplus$ is a (topological) direct sum. (Recall that J is the set of indices of the Π_m unisolvent subset of the information.) Then X_0 is a Hilbert space with $\Theta(\cdot)$ as a norm (not a semi-norm). Then given the reproducing kernel $K_M(s,t;x,y)$ of X_0, (which can be expressed in terms of the reproducing kernel $K((s,t);(x,y))$ of F_1 and the functions $q_j(x,y)$) the interpolating spline is given by :

$$\sigma_2(x,y) = \sum_{i \in J} \alpha_i \cdot K_M((x_i,y_i);(x,y)) + \sum_{j \in J} z_j \cdot q_j(x_j,y_j) \tag{4.4}$$

where the size of J is equal to d,the cardinality of the nullspace and the coefficients α_i can be calculated from the (k-d) by (k-d) dense linear system given by:

$$\sum_{i \in J} \alpha_i \cdot K_M(x_k,y_k;x_i,y_i) = z_i - \sum_{j \in J} z_j \cdot q_j(x_j,y_j) \quad \forall k \in J. \tag{4.5}$$

For a more complete treatment, including examples of the actual systems arising in practice, see Boult (1985a) or Boult (1986). For a thorough comparision of this method with that of Grimson (1981) see Boult (1985b).

Note that although equations (4.1) and (4.4) seem different, if the class of functions and the norm are the same, then the resulting splines are exactly the same function! This follows directly from the uniqueness of the function minimizing the functional $\Theta(\cdot)$ in the class F_1.

§4.3 Examples in Different Situations.

In this section we shall look at which representation (4.1) or (4.4) is best suited to imaging situations. The methods lend themselves to different implementations depending of the imaging conditions and user priorities. We present six different imaging situtations and examine what implementation details we might use to solve the resulting linear system in these situations.

§4.3.1 Data on a Large Fixed Grid: Speed Most Important.

In this example we assume the user has the means to gather samples of the surface depth on a regular grid pattern (e.g. with an accurate laser rangefinder). If the user is interested in pure speed then the representation (4.1) is better suited to the problem. Then the linear system ((4.2) or (4.3)) can be inverted (or decomposed) as a precomputation and stored. (This costs time proportional to k^3, and space proportional to k^2). When the data comes in from the rangefinder, the user simply forms a vector dot product (time n^2) to recover the spline coefficients used in (4.1).

§4.3.2 Data on a Large Fixed Grid:
Space Limited.

In this example we assume the user has samples of the surface depth on a regular grid pattern plus a few extra points (again maybe from an accurate laser rangefinder). We assume the user has limited memory and cannot afford to store and retrieve the inverse of a k by k linear system. In this case the better representation is (4.4); we need the positive definiteness and block toeplitz structure of a component of the linear system (4.5). We can then use a result of Lee (1985) that uses a conjugate gradient iterative approximation to the coefficients in time at most $O(n^2 \log n)$ and space $O(n)$. This algorithm takes advantage of the block toeplitz structure in the component of the system (using FFT's) to do a vector matrix dot product in time $O(n \log n)$ instead of $O(n^2)$.

§4.3.3 Stereo Data Gathered in Parallel:
No Update in the Location of Data
and Accuracy Most Important.

In this example we assume that we are given all the depth data at one time, and we will not wish to change the location of any of the data (though we may wish to change the value of the data if we change a stereo match). Here we are interested in achieving the most accurate solution. We would use (4.2) as our linear system (it has a substantially lower condition number), and solve it with Gaussian elimination with full pivoting (time $(1/3\ n^2)$) and a few steps of iterative refinement (cost n^2 per step). This also has the advantage that the kernels are simple, and so is the actual program to solve the linear system.

§4.3.4 Stereo Data Gathered in Parallel:
No Update in the Location of Data,
and Speed Most Important.

In this example we assume that we are given all the depth data at one time, and will not wish to change the location of any of the data although we may wish to change the *value* of the data, say if we change a stereo match. Here we are interested in achieving the fastest overall solution, and the choice is not so clear. We can use representation (4.4) and solve the system (using standard Cholesky factorization) in time $1/6\ n^3$ but have to pay a larger constant factor for each surface point: the kernel is significantly more expensive to evaluate. Or we could use the simpler kernels and factor the matrix (see below), and then using a more complicated algorithm also solve the system in $1/6\ n^3$ (with a much larger constant in the terms of order n^2). To do the latter we would factor the matrix as follows:

$$A \equiv \begin{bmatrix} Q & C \\ C^T & W \end{bmatrix} \quad \text{where Q is 2d by 2d and W is (n-2d) by (n-2d).}$$

$$= \begin{bmatrix} I_{2d} & 0 \\ C^T Q^{-1} & I_{n-2d} \end{bmatrix} \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} I_{2d} & Q^{-1}C \\ 0 & I_{n-2d} \end{bmatrix}$$

$$\text{where } R = W - C^T Q^{-1} C$$

Then we can use Cholesky factorization on R (this is because all the negative eigenvalues are in Q, so R must be positive definite as required for the Cholesky algorithm). This takes time $1/6\ n^3$. We then have three triangular systems which can be back-solved to arrive at the answer. Since the size of Q is fixed (it is 2d, where d is the dimension of the null space of the semi-norm being used), all the calculations with Q and Q^{-1} can be done in constant time. Note that the factorization does increase the condition number of the system (4.2), but only by a constant factor. See Boult (1986) for more details.

§4.3.5 Stereo Data Gathered Sequentially:
Adding Data and Removing Last Added Point
the Only Allowable Updates,
and Speed is Most Important.

In this example we assume that we are given a subset of the data all at once, and we may wish to add points to the information and at other times remove the last point we added (although we can change the value of any data point at any time). Here we are again faced with the same choices as in (4.3.4). In fact we use the same algorithms with the exception that we make provisions to add and delete the final row and column from the Cholesky decomposition (and in the factorization described in 4.3.4 we must also update the matrix C and C^T). The resulting algorithms still have cost $1/6\ n^3$ but with the added cost of $O(k^2)$ for the deletions.

§4.3.6 Stereo Data Gathered Sequentially:
Adding Data and Removing Any Data
and Accuracy Most Important.

In this example we assume that we are given a subset of the data all at once, and we may wish to add points to the information and remove arbitrary points; also, we can change the value of any data point at any time. We desire to

find the most accurate solution. In this case we would use representation (4.1) and solve the system (4.3) using a stable QR decomposition with updating (see Daniel et al (1976)), which costs about 3 j^2 for each addition or deletion (assuming there are j points in the current representation) for a total cost of about (3/2) k^3 for k data points assuming the number of deletions is a constant.

§5 Future plans.

Much of the work describe in this paper is currently under investigation. In particular, the psychology experiments described in section 2 are under way, and shall be reported on in Boult (1986). As part of the experiments we shall implement the optimal algorithms for all the different classes described in section 2.4, plus numerous other classes.

The analysis and comparison of alternative realizations will continue. The more we analyze, the more new ways of realizing the solution and implementing these realizations we have found. In addition, new advances in the solution of linear systems (either in algorithm or in hardware development) can always be applied. We shall examine how the class choice affects both the numerical accuracy of the solution and the computational complexity of the resulting algorithms.

§6 References.

Allen, Peter, (1985): Object recognition using Vision and ..., Ph.d. dissertation, University of Pennsylvania.

...mur, D.W. (1974): Multivariate Spline Functions 1: Construction, properties, and Computation, *J Approximation Theory*, #12, p396-411.

Atteia, Marc. (1966): Etude de Certains Noyaux et Théories des Functions «Spline» en Analyse Numérique. Docteur Thèse, De L'Institut de Mathématiques Appliquées de Grenoble, France.

Boult, Terrance, (1985a): Reproducing Kernels for Visual Surface Interpolation, Columbia University Computer Science Department Technical Report.

Boult, Terrance, (1985b): Visual surface Interpolation: A Comparison of Two Methods., to appear in these proceedings, DARPA Image understanding Workshop, December 1985.

Boult, Terrance, (1985c): Smoothness Assumptions in Human and Machine Vision: Their Implications for Optimal Surface Interpolation, Columbia University, Computer Science Department Technical Report.

Boult, Terrance, (1986): Information Based Complexity: Applications in Nonlinear equations and Computer Vision, Doctoral dissertation, in preparation.

Daniel, J.W., W.B. Gragg, L. Kaufman, and G.W. Stewart (1976): Reorthogonalization and Stable Algorithms for Updating the Gram-Schmidt *QR* Factorization, *Math. Comp*, v28, p772-795.

Franke, R. (1984): Thin Plate Splines with Tension, to appear *CAGD*.

Grimson, W.E.L., (1981): *From Images to Surfaces: A Computational Study of the Human Early Visual System*, MIT Press, Cambridge, MA.

Yender, John, David Lee and Terrance Boult. (1985): Information Based Complexity Applied to the 2 1/2 D Sketch, *Proceedings of the Third IEEE Workshop on Computer Vision: Representation and Control*, p157-167.

Marr, David (1977): Artifical Intelligence – A personal View, *Artifical Inteligence 9*.

Mayhew, John, (1982): The Interpretation of Setreo Disparity Information: The computation of Surface Orientation and Depth, *Perception*, vol 11, p.387-403.

Meinguet, Jean, (1979a): Multivariate Interpolation at Arbitrary Points Made Simple, *Journal of Applied Mathematics and Physics 30*, 292-304.

Meinguet, Jean, (1979b): Basic Mathematical Aspects of Surface Spline Interpolation, *ISNM 45: Numerische Integration*, 211-220, G. Hämmerlin ed., Basel: Birkhäuser Verlag.

Terzopoulos, Demetri, (1984): Multiresolution Computation of Visual Surface Representation, Ph.d. thesis, MIT.

Traub, J.T. and H. Wozniakowski, (1980): *A General Theory of Optimal Algorithms*, Academic Press NY.

Traub, J.T., G. Wasilkowski, and H. Wozniakowski, (1983): *Information, Uncertainty and Complexity*, Addison Wesley, MA.

GC970145-86

LABELLING LINE DRAWINGS OF CURVED OBJECTS

Jitendra Malik

Computer Science Department
Stanford University, Stanford, California 94305

Abstract

Understanding how a line drawing is interpreted as the projection of a set of three dimensional objects is a problem of fundamental importance in Computational Vision. This can be viewed as a two step process (1) Labeling the line drawing and (2) Inferring surface constraints from the labelling by further geometric reasoning.

This paper describes an algorithm for the labelling problem for line drawings of opaque, curved objects bounded by piecewise smooth surfaces. Each image curve is classified as the projection of a limb the locus of points on the surface where the line of sight is tangential to the surface or as an edge a tangent plane discontinuity. Additionally each edge is classified as convex, concave or occluding. The scheme is mathematically rigorous and is complete for scenes with no surface markings or shadows.

1 Introduction

Line drawing interpretation is one of the most important modules in visual perception. A line in a drawing can correspond in the scene to a discontinuity in depth, surface orientation, surface reflectance or illumination. We'll however deal with a simplified model of the world where the objects have no surface marks on them and where the lines due to illumination discontinuities like shadow edges and specularities has been removed in some preprocessing step. The local structure of the intensity surface can be used to do this [23]. Each line (image curve) can then be classified as either the projection of a *limb* the locus of points on the surface where the line of sight is tangent to the surface or as an *edge* a tangent plane discontinuity. Additionally each edge can be classified as convex, concave or occluding edge. For occluding edges and limbs we would like to infer which of the two surfaces bordering the curve in the line drawing is nearer in the scene. These inferences can be represented by giving each line one of 6 possible *labels*.

1. A "+" label represents a convex edge an orientation discontinuity such that the two faces meeting along the edge in the scene enclose a dihedral angle greater than π.
2. A "-" label represents a concave edge an orientation discontinuity such that the two faces meeting along the edge in the scene enclose a dihedral angle less than π.
3. A "←" or a "→" represents an occluding convex edge. When viewed from the camera, both the surface patches which meet along the edge lie on the same side, one occluding the other. As one moves in the direction of the arrow, these surfaces are to the right.
4. A "←←" or a "→→" represents a limb. Here the surface smoothly curves around to occlude itself. As one moves in the direction of the twin arrows, the surface lies to the right.

Of the 6^n combinatorially possible label assignments to the n lines in a drawing only a small number are physically possible. The determination of these is the the *labelling* problem and is the primary subject of this paper.

2 Review of past work

The first successful attempt to solve this problem was made by Huffman [7] and Clowes [3] in 1971. They exhaustively cataloged the vertices that could arise in line drawings of trihedral objects (objects whose corners are formed by exactly three meeting faces) and then used the catalog to interpret lines as corresponding to convex, concave or convex occluding edges. The catalog gives the possible labellings at each junction and global consistency is forced by the rule that each line in the drawing be assigned one and only one label along its length. Waltz [20] proposed an algorithm for this problem (actually for an augmented version with shadows, cracks, and separably concave edges) which reduced the search by a filtering step in which adjacent pairs of junctions are examined and incompatible candidate labellings discarded. Mackworth [12] developed the concept of Gradient Space which enabled his program to label line drawings of arbitrary polyhedral scenes. One consequence of this attempt to deal with an arbitrary number of planes meeting at a vertex was a combinatorial explosion

in the number of labellings generated which correspond to highly counter intuitive interpretations. Draper [5] points out that there are 33 legal labellings(96 if accidental viewpoint is allowed) for the line drawing of a tetrahedron. In the context of the Origami World Kanade [8] had to face a similar problem.

For objects bounded by curved surfaces, there have been two major efforts. The first was by Turner [19] who used a heuristic procedure called the PC (polyhedral to curved) transformation to obtain the labelling possibilities for junctions. Turner's approach suffered from several basic weaknesses:

- Unlike the Huffman-Clowes procedure which was obviously complete for the class being considered, such a convincing claim cannot be made for Turner's procedure.
- Turner's procedure is limited to objects such that each face is only one type of surface–planar, parabolic but non planar, elliptic or hyperbolic. A simple object like a torus which is bound by a single smooth surface which has both elliptic and hyperbolic patches can not be handled.
- A major problem is the huge number of junction labels and the consequent explosion in the number of legal interpretations. See Table 1. This is to be contrasted with the small size of the Huffman Clowes catalog.

The next major effort was by Shapira-Freeman-Chakravarty [17], [4]. They considered objects such that exactly three faces meet at a vertex where each face is either a quadric surface or a plane. Their junction catalog is much smaller than that of Turner which makes it practically usable in certain situations. However some fundamental weaknesses remain:

1. No arguments are given to prove the validity of the junction catalog. One is left with the suspicion that it was 'derived' by observation of junctions in some typical curved objects.
2. The scheme is limited to objects bounded by quadric surfaces/planes and exactly three faces meeting along a vertex.
3. For a non-occluding edge, no distinction is made between convex and concave edges.

Corner	Labels
P_3	152
P_4	652
C_1	16
C_2	348
C_1P_1	138
C_1P_2	1905
E_1	8
E_2	205
E_1P_1	138
E_1P_2	443

Table 1 Number of labellings for each of Turner's corner classes

Lee, Haralick and Zhang [11] extend this catalog by adding line labels based on Huffman-Clowes rules. The justification for the validity of this step is not given. While this partially solves problem 3 mentioned above the first two weaknesses remain. For a detailed discussion we would refer the reader to [15] where we also point out some mistakes in these catalogs.

3 Criteria for evaluating a labelling scheme

We list here some 'self-evident' criteria which one would like a scheme for labelling curved objects to satisfy.

1. It should be able to handle a broad range of objects – n faces meeting at a vertex with each face a portion of a general surface. A restriction to planes or quadric surfaces is too limiting.
2. The scheme should be derived in a mathematically rigorous way. It should be possible to state the precise conditions under which the scheme is valid.
3. The algorithm should not generate too many labellings. Preferably they should correspond to the interpretations found by human observers.
4. The labellings generated should be physically realizable.
5. Labellings should be found without too much search.
6. The scheme should be robust with respect to errors in the input line drawing.

In Section 11, we will evaluate our scheme for its success (or failure !) in meeting these criteria.

4 Modelling the Scene and the Projection

The scene consists of a set of *objects* in three dimensional space. An *object* is a connected, bounded and regular subset of $\mathbf{R}^3$ whose boundary is a C^3 piecewise smooth surface. By *regular* we mean that it is the closure of its interior. This disallows objects with "dangling" faces or edges. Imposing this condition is a standard practice in solid modelling. The formal definition of *piecewise smooth surface* is somewhat complicated and the interested reader is referred to [15] for the details. Basically a piecewise smooth surface consists of portions of smooth surfaces joined together. The surface of a polyhedron or a finite cylinder are simple examples. If two C^3 surfaces $F(x,y,z) = 0$ and $G(x,y,z) = 0$ intersect at a point P where the surfaces have distinct tangent planes, then it can be shown that the part of the intersection of the two surfaces near P is a smooth C^3 arc. Such an arc on a piecewise smooth surface is called an *edge*. A point of intersection of three or more

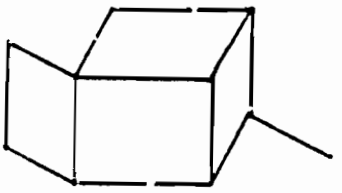


Figure 1: Not an Object

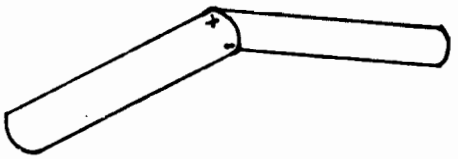


Figure 2: Not a piecewise smooth surface

edges is called a ***vertex***. Each maximal connected smooth portion of the surface is called a ***face***. These definitions reduce to the standard definitions of faces, edges and vertices for polyhedra.

Example. A right circular cylinder is a C^∞ piecewise smooth surface with 3 faces, 2 edges and 0 vertices.

Example. A sphere or a torus is a piecewise smooth surface with 1 face, 0 edges, and 0 vertices.

We would like to point out that our definition of piecewise smooth surfaces differs somewhat from some other definitions, which do not require that the tangent planes be distinct across an edge. An example of a surface which is *not* piecewise smooth under our definition is shown in Figure 2. The cone, if its apex is included is also not a piecewise smooth surface.

Two kinds of mappings may be considered orthographic and perspective. In this paper we'll limit ourselves to orthographic projection which corresponds to the eye/camera being effectively at infinite distance from the scene. If the viewing direction is along the z axis, a point (x,y,z) in the scene projects to the point (x, y) in the image plane and is visible if there is no other point $(x\ y,\ z')$ belonging to any object in the scene with $z' < z$. The projection of the scene is the projection of the visible points in the scene.

The viewpoint is assumed to be ***general*** - there exists an open neighborhood of the vantage point in which the 'topological' structure of the line drawing remains unaltered. This will be made more precise in the next section.

5 The Line Drawing

The only lines we will be concerned with are the projections of depth and orientation discontinuities in the scene. The curves in the line drawing are segmented at tangent/curvature discontinuities. The points where there are tangent/curvature discontinuities are referred to as junctions. Figure 3 is a sample line drawing. Endings of image curves e.g. j_3 are also referred to as junctions. Js are points where two or more image curve segments meet e.g. j_1, j_5, j_6.

From the line drawing an image structure graph may be constructed. It is an undirected graph. Its nodes are all the junctions in the line drawing and additionally *pseudo* junctions like j_8 one for each isolated smooth closed curve like C_1. Each image curve segment corresponds to an arc between the vertices corresponding to the junction/s on which it is incident. for example the node corresponding to j_4 has arcs corresponding to C_2,C_3,C_4 the first two to the node corresponding to j_7 and the third to the node corresponding to j_2. The node corresponding to j_6 has three arcs, the ones corresponding to j_2-j_8 have only one arc each. As is obvious, in general the Image Structure Graph (henceforth the ISG) may be disconnected, have more than one arc between two nodes and have self loops.

By considering the observed geometric properties in the line drawing these junctions can be classified as follows:

Termination: Curve ends there e.g. j_3.

L: Tangent discontinuity across junction. e.g. j_1.

Curvature L: Tangents continuous, Curvature discontinuity e.g. j_2

T- Junction: 2 of 3 image curves at the junction have same slope, curvature e.g. j_6.

Pseudo: Corresponds to isolated closed smooth curves e.g. j_8.

Three-Tangent: 3 curves with common tangent. 2 have same curvature e.g. j_4.

Arrow: 3 curves with distinct tangents. One angle $> \pi$ e.g. j_9.

Y: 3 curves with distinct tangent's. No angle $> \pi$ e.g. j_5.

Multi: 4 or more image curves at the junction.

The ISG is augmented by storing at each node an attribute field corresponding to which of the above classes the junction belongs.

We can now define precisely what we mean by general viewpoint. The augmented ISGs corresponding to the line drawings of the scene when viewed from different points of a sufficiently small open neighborhood (a ball) of the vantage point are isomorphic.

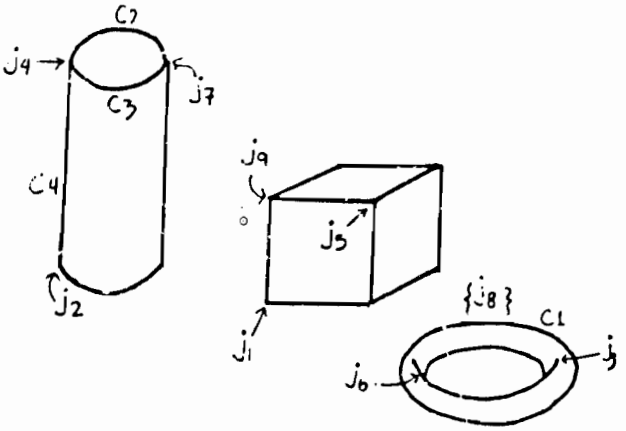


Figure 3: A Sample Line Drawing

6 Projection of curved objects

We wish to study how curved objects project to a line drawing. This is done by studying how neighborhoods of different kinds of points on the surface project and cataloging the resulting junctions in the line drawing. This can be broken up into three cases.

- The projection of a neighborhood of an interior point of a face.
- The projection of a neighborhood of an interior point of an edge.
- The projection of a neighborhood of a vertex.

which are tackled respectively in Sections 7, 8 and 9.

As we are dealing with the projection of *opaque* surfaces, we also have to worry about the phenomenon of occlusion obstruction of the view of the surface of an object by another object (or another part of the same object). This gives rise to T junctions. Here we know that the top of the T junction corresponds to a nearer surface occluding another object. See Figure 4. Note that there is no constraint on the label of the stem of the T junction.

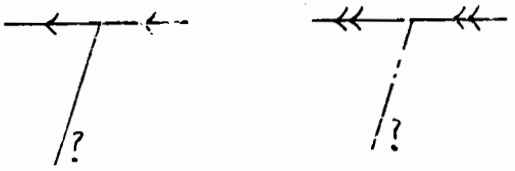


Figure 4: Labels for a 'T' Junction

7 Projection of a face neighborhood

This corresponds to the projection of a single C^3 surface element. it is an instance of the class of mappings from two dimensional manifolds to two dimensional manifolds. Whitney in 1955 studied the singularities of such mappings and showed that generically there are only two singularities the *fold* and the *cusp*. This result is discussed in Section 7.1. In section 7.2 we study Whitney's theorem in the context of the projection mapping. Limbs are associated with the fold singularity and terminations(Section 4) are associated with cusps. At a termination, we can determine which of the two surface patches is nearer.

7.1 Whitney's Singularity Theory

In 1955, Whitney [21] published a landmark paper on the singularities of mapping of open sets in E^2 into E^2. Examples of such mappings are projection (orthographic and perspective) of surfaces and the Gauss map. Whitney showed that generically there are only two kinds of singularities: *folds* lying along curves and isolated *cusp* points lying on the folds. We will explain what this means in the context of projection in the next section. In is section we'll define various terms and try to give a feel for Whitney's results.

Such a mapping is defined by the two functions $u_1 = f(x_1, x_2)$ and $u_2 = (x_1, x_2)$ where (x_1, x_2) and (u_1, u_2) are the coordinates in the two spaces. The mapping is said to be C^k if the functions f and g have continuous partial derivatives of order $\leq k$. Let J be the Jacobian of the mapping. A point p is said to be a regular or singular point according as $J(p) \neq 0$ or $J(p) = 0$. We are interested in studying the locus of the singularities.

Example 1. Consider the mapping

$$u_1 = {x_1}^2, u_2 = x_2$$

The Jacobian $J = 2x_1 = 0 \Rightarrow x_1 = 0$ The straight line $x_1 = 0$ is the locus of points where the mapping is singular. This is an example of a fold.

Example 2. Consider the mapping

$$u_1 = {x_1}^3 - x_1 x_2, u_2 = x_2$$

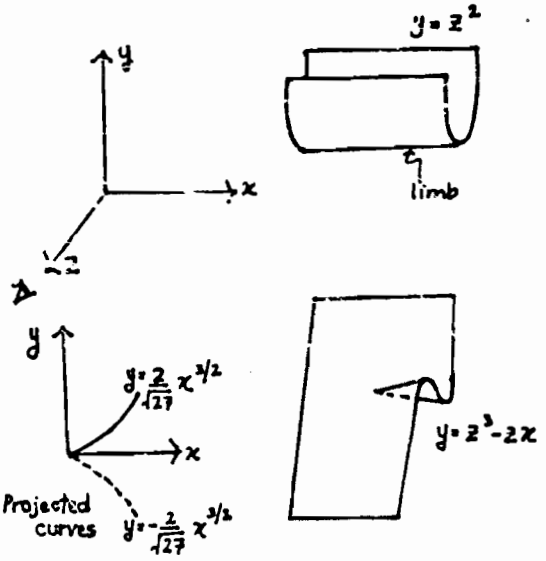


Figure 5: Two canonical examples

The Jacobian $J = 3x_1^2 - x_2 = 0 \Longrightarrow x_2 = 3x_1^2$. This corresponds to two fold curves, one for positive x_1 and one for negative x_1— both in the half plane where $x_2 \geq 0$. The two fold curves meet at the origin at a cusp point. This singularity occurs whenever two fold curves come together and disappear.

Our choice of examples was not accidental. After suitable coordinate transformations all folds and cusps can be described by the canonical forms in Examples 1 and 2 respectively. Whitney showed that every singularity of a smooth mapping from E^2 to E^2, after an appropriate small deformation splits into folds and cusps. As we discuss in the next section this *generic* property in the vision context corresponds to general viewpoint.

7.2 Singularities of the Projection mapping

Projection is a mapping from a surface onto a plane. One can immediately interpret Examples 1 and 2 as corresponding to the orthographic projection of two surfaces of the form $y = f(x, z)$ viewed from an infinitely distant point on the z-axis. Figure 6 shows the two surfaces. For $y = z^2$ the projection of the fold curve is the line $y = 0$.

For $y = z^3 - zx$ the two fold curves have the equation

$$x = 3z^2$$

one for z positive, other for z negative.

By eliminating z we get the equations of the projected curves

$$y = z^3 - z(3z^2) = -2z^3 = -2(\pm\sqrt{x/3})^3 = \pm\frac{2}{\sqrt{27}}x^{3/2}$$

This is a semi cubic parabola with a cusp at the origin. Only the positive branch is visible. Note that the contour ends concavely. For an extended discussion of this see [10]. This fact can be used to determine to which side the curve belongs.

What does all this imply for the labelling problem? The only curves which exist in the projection of a smooth surface patch are limbs, and the only junctions are terminations. In the line drawing each limb projection borders two regions, or sometimes two strips of the same region. The limb curve lies on the surface patch corresponding to one of these strips and is in front of the other surface patch. Which is the nearer patch can be determined by looking at the curvature of the projection of the limb(in the image plane) at a termination junction as shown in Figure 6. If the scene consists only of objects bound by single smooth surfaces(no edges), then the only junctions in the line drawing would be T junctions and terminations.

8 Projection of an edge neighborhood

An edge e is the intersection of two surface patches S_1 and S_2 with different tangent planes. Consider a point P in the interior of this edge. To study how P and its neighbourhood in S_1, S_2 project we have to consider three cases:

- No limb through P on either S_1, S_2.
- Limb through P on both of S_1, S_2.
- Limb through P on one of S_1, S_2.

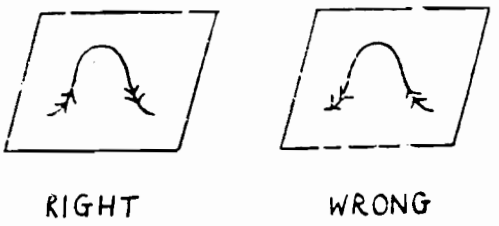


Figure 6: Inferring the labelling from terminations

The first case is easy. Since there is no limb on either patch, the projection of both the patches are diffeomorphisms. The edge segment in the neighborhood of P is the boundary of both patches and hence the boundary of both their projections. The labels on the edge on both sides of P are the same.

The second case is also easy. As there is a limb on S_1, the ray from the viewer must lie on the tangent plane to S_1 at P. Similarly it must lie onn the tangent plane to S_2 at P. It therefore must lie on their intersection which is a straight line. In other words, the vantage point is constrained to lie along a line, which violates the general viewpoint assumption.

The third case is more interesting. Without loss of generality we can assume that S_1 has a limb passing through P. For S_2 we will assume a general equation and then do the case analysis. Cartesian coordinates are introduced with the origin at P. The eye is at infinite distance along the z axis, so that the projection is on the x y plane. See Figure 7.

For surface S_1

$$y_1 = a_2x^2 + a_3zx + a_4z^2$$

We are using Whitney's result on the normal form of the limb. As for us the direction of projection is fixed we do not have the freedom to get rid of the rotation term.

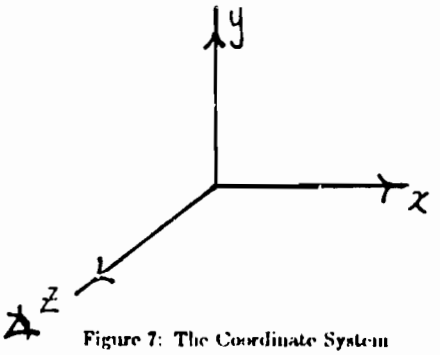


Figure 7: The Coordinate System

Now, the limb is given by

$$\frac{\partial y_1}{\partial z} = 2a_2z + a_3x = 0 \Rightarrow z = \frac{-a_3x}{2a_2}$$

By substituting back we get

$$y_1 = (a_4 - \frac{a_3^2}{4a_2})x^2$$

This is the equation of the limb on S_1 in the neighborhood of the origin.

For surface S_2

$$y_2 = b_0 + b_1 x + b_2 z^2 + b_3 zx + b_4 x^2$$

As before, any limb on this would be given by

$$\frac{\partial y_2}{\partial z} = b_0 + 2b_2 z + b_3 x = 0 \Rightarrow z = \frac{-b_3 x - b_0}{2b_2}$$

By substituting back we get the limb equation

$$y_2 = -\frac{b_0^2}{4b_2} + (b_1 - \frac{b_0 b_3}{2b_2})x + (b_4 - \frac{b_3^2}{4b_2})x^2$$

Recall that we have assumed that there is no limb on this patch passing through P. This implies that $b_0 \neq 0$.

The equation of the intersection curve/edge is given by

$$-b_0 z - b_1 x + (a_2 - b_2)z^2 + (a_3 - b_3)zx + (a_4 - b_4)x^2 = 0$$

from which

$$\frac{dz}{dx} = \frac{b_1 - 2(a_4 - b_4)x - (a_3 - b_3)z}{-b_0 + 2(a_2 - b_2)z + (a_3 - b_3)x}$$

What we really want is the slope of the tangent to the projected edge ie

$$\frac{dy}{dx} = \frac{\partial y}{\partial z}\frac{dz}{dx} + \frac{\partial y}{\partial x}$$

Substituting we get,

$$\frac{dy}{dx} = (2a_2 z + a_3 x)\frac{b_1 - 2(a_4 - b_4)x - (a_3 - b_3)z}{-b_0 + 2(a_2 - b_2)z + (a_3 - b_3)x} + (a_3 z + 2a_4 x)$$

As $b_0 \neq 0$, $\frac{dy}{dx} = 0$ at the origin which means that the projection of the intersection curve is tangent to the projection of the limb curve.

We are now ready to list the junctions which arise from the projection of a sufficiently small neighborhood of P. Note that S_1 in the neighborhood of P is a cylindrical patch (This is from Whitney's theorem by suitable change of coordinates the surface along a fold curve can be described by $z = x^2$) Let TP_1 be the tangent plane to S_1 at P. The viewpoint is constrained to be in this plane. Let TP_2 be the tangent plane to S_2 at P. S_1 and S_2 divide the three dimensional space in the neighborhood of P into four quadrants as shown in Figure 8. By putting in solid material in various quadrants and viewing from various directions in TP_1 we can generate all possibilities. The reader will note the similarity of this procedure to the Huffman Clowes procedure described in [7] To begin the case analysis:

1. Solid material in only one quadrant. This gives two subcases

   (a) Solid material in quadrant 1. Depending on whether the viewpoint is in the upper or lower half plane we get the two junctions in Figure 9. Recall that we had shown earlier that the projection of the edge is tangent to the projection of the limb.

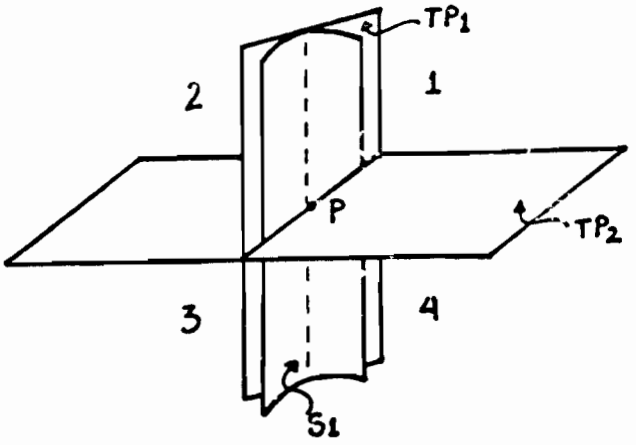


Figure 8: The four quadrants

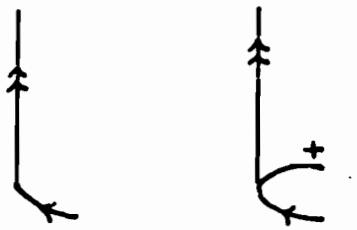


Figure 9: Viewing quadrant 1.

(b) Solid material in quadrant 2. This gives us the 'junction' in Figure 10. Note that here the limb curve itself is occluded. Unlike the other junctions, this cannot be identified because it does not correspond to any tangent/curvature discontinuity in the line drawing. We have to allow for the possibility of this junction being present by introducing 'phantom' nodes on all curved lines in the drawing which could correspond to edges. (This 'junction' cannot occur on projections of limbs or concave edges)

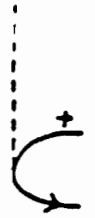


Figure 10: Another 'junction'

Figure 11: One more labelling for a Curvature-L

2. Solid material in two quadrants. There are two subcases. One is when adjacent quadrants are occupied. In that case there is only one surface at P. The other case is when opposite quadrants are occupied, for example (1,3) or (2,4). In this case, e violates our definition of an edge.

3. Solid material in three quadrants. In the first subcase, let 1 be the empty quadrant. In this case P is hidden. If 2 is the empty quadrant, we get the junction in Figure 11.

4. Solid material in all four quadrants. Here no surfaces are defined.

9 Projection of a vertex neighborhood

We begin our analysis of the projection of a vertex with the observation that under general viewpoint no limb can pass through a vertex. As the vantage point moves the limb curve moves on the surface and it intersects the vertex only for a particular viewpoint. Our next observation is the following result

Theorem. If two surface elements S_1 and S_2 intersect along a curve C at the point P, then TP_1 — the tangent plane to S_1 at P and TP_2 — the tangent plane to S_2 at P intersect along a straight line L such that at P the projection of L on the image plane is *tangent* to the projection of C.

Proof. The coordinate system is the same as in the previous section. Consider the Taylor Series Expansions of the graphs of the two surfaces with P as the origin. The 0'th and higher order terms can be ignored without loss of generality.

$$y_1 = a_0z + a_1x + a_2z^2 + a_3zx + a_4x^2$$

$$y_2 = b_0z + b_1x + b_2z^2 + b_3zx + b_4x^2$$

First let us find the equation of the edge curve along which these surfaces intersect. This is given by

$$(a_0 - b_0)z + (a_1 - b_1)x + (a_2 - b_2)z^2 + (a_3 - b_3)zx + (a_4 - b_4)x^2 = 0$$

from which

$$\frac{dz}{dx} = -\frac{(a_1 - b_1) + 2(a_4 - b_4)x + (a_3 - b_3)z}{(a_0 - b_0) + 2(a_2 - b_2)z + (a_3 - b_3)x}$$

What we really want is the slope of the tangent to the projected curve i.e.

$$\frac{dy}{dx} = \frac{\partial y}{\partial z}\frac{dz}{dx} + \frac{\partial y}{\partial x}$$

Substituting we get,

$$\frac{dy}{dx} = -(a_0 + 2a_2z + a_3x)\frac{(a_1 - b_1) + 2(a_4 - b_4)x + (a_3 - b_3)z}{(a_0 - b_0) + 2(a_2 - b_2)z + (a_3 - b_3)x} + (a_1 + a_3z + 2a_4x)$$

At the origin, this simplifies to

$$\frac{dy}{dx} = \frac{a_0b_1 - a_1b_0}{a_0 - b_0}$$

Now consider the equations of the two tangent planes

$$y_1 = a_0z + a_1x$$

$$y_2 = b_0z + b_1x$$

The equation of the intersecting line is given by

$$(a_0 - b_0)z + (a_1 - b_1)x = 0$$

Using this to eliminate z we get

$$y = \frac{a_0b_1 - a_1b_0}{a_0 - b_0}x$$

which is the same slope as required.

This result has an immediate consequence. The projection of a vertex *locally* looks like the projection of an equivalent polyhedral vertex formed by replacing each of the surface elements by their tangent planes. This results in a great simplification in the analysis, as all the results on polyhedral junction labelling discussed in Section 2 become relevant. Find the equivalent straight line junction by replacing each image curve at the junction by its tangent and look up (or derive) the labelling possibilities from a polyhedral junction catalog. For example if it is known a priori that exactly three surface elements meet at a vertex, then the labelling possibilities are exactly those of the Huffman Clowes set. As we want to deal with a more general class, further analysis will be necessary as in the next subsection.

9.1 Labelling Polyhedral Junctions

In Section 2, we reviewed perhaps the two most significant pieces of work on this problem — starting with Huffman Clowes labelling for the trihedral world and Mackworth's Gradient Space approach for dealing with arbitrary number of surfaces meeting at a vertex. We also noted Draper's empirical observations on the combinatorial explosion in the number of alternative interpretations when no restrictions are made on the number of surfaces meeting at a vertex.

It is clear that we need some way of pruning these weird interpretations. Kanade in [8] exploited with great success the fact that parallel lines in the scene project to parallel lines in the image and symmetries project to skewed symmetries. These heuristics unfortunately are useful only for the classes of objects which have parallel edges and faces with axes of symmetry. We need a criterion which is more generally applicable. Several attempts [1, 3] have been made to find simplicity criteria/minimization schemes to find the psychologically preferred interpretations of line drawings. Most of these approaches are limited to isolated image curves. Instead of attempting to find grand global simplicity criteria we limited ourselves to local simplicity. Based on the

observation that all the highly counterintuitive interpretation involve a number of hidden faces we develop our criterion *for each junction find the vertex interpretations with the minimum number of faces meeting at the vertex*. The interpretations should be stable under general viewpoint. Now for polyhedra at a vertex there are exactly two faces sharing an edge, and exactly two edges bounding a face, it follows that the number of edges at a vertex is equal to the number of faces incident at the vertex. Therefore we have an equivalent version of the rule find interpretations involving minimum number of edges.

An example will make this clearer. Consider an arrow junction. Each of the three lines which meet at the junction is the projection of an edge which is incident on the corresponding vertex. From our simplicity rule, we try to find vertex interpretations which require only 3 edges or equivalently only three faces meeting at the vertex. This means that for the arrow, our local labeling possibilities are the same as that for the Huffman Clowes scheme for the trihedral world. By the same reasoning, for the Y junction again we get the same three labelling possibilities as in the Huffman Clowes scheme. For L-junctions we need three faces as well there are no legal interpretations with two faces. Here again, the labeling possibilities are the same as that of the Huffman Clowes scheme.

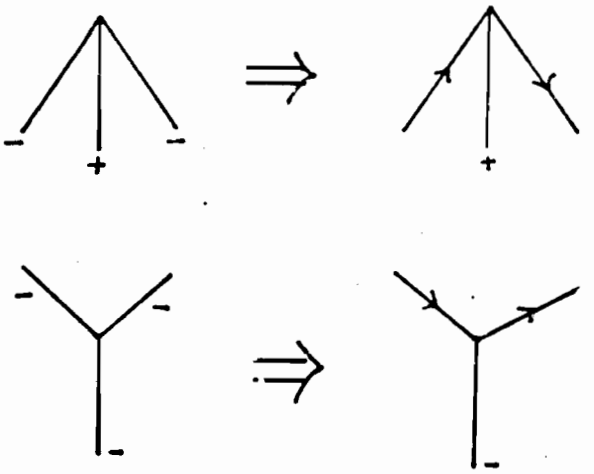


Figure 12: Generating junctions with occlude labels

For higher-order junctions with 4 and more line meeting at the junction we need n faces meeting at the vertex if there are n lines meeting at the junction. We need to develop an algorithm for generating the labelling possibilities for such higher order junctions as may occur in a line drawing eg in overhead views of square pyramids. We will first need an auxiliary result proved in [15]

Theorem. If there are n lines meeting at a junction, $n \geq 3$, all the labelling possibilities for the junction which correspond to minimum number of faces(n) at the vertex, correspond to either n,2 or one hidden face at the junction.

Now we are ready to generate the labelling possibilities. To do this, first consider the simpler case all the labels are convex or concave. This corresponds to the case when there is no hidden face. If a labelling is legal it should be possible to construct a reciprocal figure in gradient space.

Instead of treating this as a geometric construction with ruler and pencil, we can use vector notation. Let $u_1, u_2, \ldots, u_n$ be the (outward) pointing vectors corresponding to the lines which meet at the junction. Let $v_1, v_2, \ldots, v_n$ be unit vectors perpendicular to these lines corresponding to a counter-clockwise rotation by 90 degrees. Consider the vector equation

$$l_1 v_1 + l_2 v_2 + \cdots + l_n v_n = 0$$

where $l_1, l_2, \ldots, l_n$ are n scalars. We will refer to this equation as the Fundamental Junction Equation. If the reciprocal figure is constructible, it corresponds to a solution of this equation. Note that our *n faces for n lines* assumption is implicitly buried in this equation. If there were hidden lines corresponding to two hidden faces meeting, they would have given rise to additional terms on the left hand side of the equation.

This equation being a 2 D vector equation is actually two linear equations in the n unknowns $l_1, l_2, \ldots, l_n$. A convex(concave) labelling for a line implies that the corresponding variable $l_i > 0$ (< 0). The set of labelling for the lines meeting at the junction corresponds to a set of constraints on these unknowns. If the system of linear equations and inequalities has a feasible solution, the labelling is legal, else it is not. The naive approach for doing this would be to solve 2^n Linear Programming problems in order to verify all possible legal labellings. One can do much better than that - a fast algorithm is described in [15].

By changing the signs of all the variables in the linear system described above we see that labellings come in pairs. This is another way of explaining the Necker ambiguity corresponding to the convex/concave reversal.

To find the legal labellings corresponding to one hidden face is easy. Take a legal all (convex/concave) edge labelling. Each pair of adjacent lines at the junction define a sector. Consider a sector defined by two lines A and B which have been labelled concave. Consider the face defined by the corresponding edges. If this face were hidden as both these lines corresponded to convex occluding edges instead of concave edges, the reciprocal figure would remain the same. One can therefore label A and B as occluding convex edges with the direction of the arrow such that the sector defined by A and B is to the left. Figure 13 shows this procedure.

This hierarchical determination of labelling possibilities, first between convex and everything else in the solution of the Fundamental Junction and then subsequent refinement is also a good strategy for doing consistency checking. For example consider arrow Y pairs. At the Fundamental Junction Equation level the arrow has only 2 labelling possibilities related by a Necker flip, the same is true for a Y. For an arrow Y path one can *without search* find the exactly 2 possible consistent labellings. See Figure 14 for a nice example.

With hind sight, it appears that Mackworth's decomposition first choosing between connect/non-connect and then for connect edges between convex/concave was not efficient from a search point of view. Similarly Waltz's approach of expanding the label set by considering shadow edges, rough orientations etc while helpful in constraining the final number of interpretations found, is a bad idea from the search point of view.

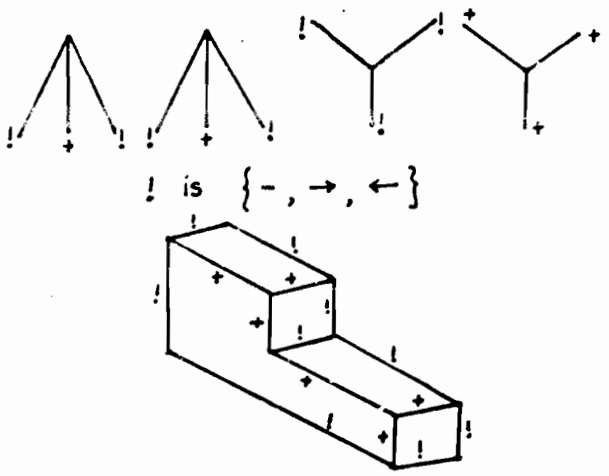


Figure 13: Using a collapsed label set to reduce search

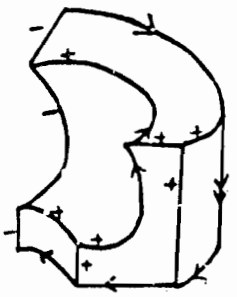
10 The labelling algorithm

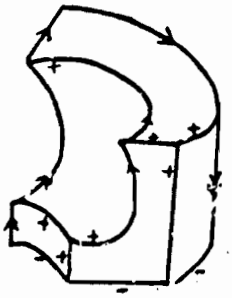
Ignoring for the moment the invisible 'junction' in Figure 10, conceptually the labelling algorithm is straightforward. The local possibilities at each junction with ≤ 3 lines have been enumerated in Sections 6–9 and for multi junctions, the labellings can be computed by the procedure described in Section 9.1. Consistency is forced by requiring the label at each end of the line to be the same.

In the previous section, we saw how by collapsing the label set, a great speedup could be obtained. That is just a special case of the notion of *relaxed* constraint satisfaction problems [15] which can lead to great speedup (in the heuristic sense) for certain kinds of problems. Line labelling being one of them. These ideas are used to develop a fast algorithm:

1. Split the ISG at T-junctions and find connected components. Each component can then be labelled independently.
2. Label the drawing with the label set { limb, edge }. For each of these labellings perform steps 3–6 below.
3. Introduce phantom nodes on arcs corresponding to curved edges.
4. Label the edges with the label set { convex, non-convex }. For each of these labellings perform steps 5–6.
5. For each junction find labelling possibilities consistent with the previously assigned coarse labels.
6. Perform a node and arc consistency filtering followed by a backtrack search to generate all labellings.

This algorithm was implemented and tested on several hand-input line drawings. It had earlier been feared that there would be a large number of interpretations due to the phantom nodes. The labellings we obtained substantially corresponded to intuitive interpretations. The most common ambiguity was between concave and occluding convex edges, corresponding to the inability to decide whether an object was stuck to a table or wall, or floating in air. Unlike what has been common in line drawing labelling work, we did not assume that there is a background such that all lines bordering it are occluding. This is merely a heuristic–it fails for holes. To take some examples, the line drawing of a cylinder has two labellings corresponding respectively to a cylinder floating in air or a cylinder resting on a table. We will close by giving the possible labellings of a curved object.





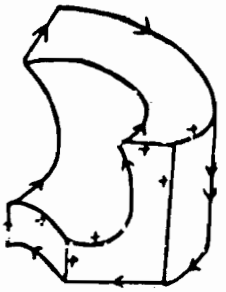


Figure 14: Possible Labellings of a Curved Object

11 Evaluation

We urge the reader to judge our performance with respect to the criteria in Section 3. In our opinion we have met criteria 1-3 and 5. On 4. our performance is good, but we do not yet have a rigorous realizability test like that of Sugihara for polyhedra. This is the subject of ongoing work. On criterion 6, our performance is poor, because the scheme relies on being able to segment at tangent/curvature discontinuities. However this scheme could be combined with the use of local image intensity information in a verification loop, making it more robust.

Acknowledgements

I wish to thank Thomas O. Binford and Christos H. Papadimitriou for many useful criticisms and discussions. This work was supported by ARPA contract N00039-84-C-0211, NASA contract NEA 80-19628 and by an IBM fellowship.

References

[1] Barrow, H.G. and J.M. Tenenbaum:, "Interpreting line drawings as three-dimensional surfaces," *Artificial Intelligence*, 17 (1981), 75-116.

[2] Binford, Thomas O., "Inferring surfaces from images," *Artificial Intelligence*, 17 (1981), 205-244.

[3] Brady, Michael and Alan Yuille, "An Extremum Principle for Shape from Contour," *Proceedings of IJCAI-8*(Karlsruhe: August 1983), 969-972

[4] Chakravarty,I., "A generalized line and junction labelling scheme with applications to scene analysis," *IEEE Trans. Pattern Anal. Machine Intelligence*1(2)(1979) 202-205

[5] Clowes, M B. "On seeing things," *Artificial Intelligence*, 2 (1971), 79-116.

[6] Draper, Stephen W., "Reasoning about depth in line-drawing interpretation," PhD thesis, Sussex University 1980.

[7] Huffman, D A., "Impossible objects as nonsense sentences," *Machine intelligence*, 6 (1971), 295-323.

[8] Kanade, T. "Recovery of the three-dimensional shape of an object from a single view," *Artificial Intelligence*, 17 (1981), 409-460.

[9] Kirousis, L. and C H Papadimitriou, " Complexity of Recognizing Polyhedral Scenes," Stanford Tech Rept 1984

[10] Koenderink, J.J. and A.J. van Doorn," The shape of smooth objects and the way contours end," *Perception* 11 (1982), 129-137.

[13] Mackworth, A.K., "Consistency in Networks of Relations," *Artificial Intelligence*, 8 (1977), 99-118.

[14] Mackworth, A.K. and E.C.Freuder, "The Complexity of Some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems," *Artificial Intelligence*, 25 (1985), 65-74.

[15] Malik, Jitendra, "Interpreting Line Drawings of Curved Objects", PhD. Thesis, Stanford University 1985.

[16] Marr, David, *Vision* (San Francisco: W.H. Freeman and Co., 1982).

[17] Shapira,Ruth and Herbert Freeman,"Computer Description of Bodies Bounded by Quadric Surfaces from a Set of Imperfect Projection," *IEEE Trans. on Computers*, September 1978

[18] Sugihara, K., "Quantitative analysis of line drawings of polyhedral scenes," *Proc. Fourth Int. Joint Conference on Pattern Recognition*, (Kyoto, 1978), 771-773.

[19] Turner,K.J.,"Computer perception of curved objects using a television camera," PhD Dissertation, Edinburgh University, Edinburgh, Scotland, 1974.

[20] Waltz, D., "Understanding line drawings of scenes with shadows," *The Psychology of Computer Vision*, Ed. P.H. Winston (McGraw-Hill, 1975).

[21] Whitney, H., "Singularities of mappings of Euclidean Spaces, I: Mappings of the plane into the plane," *Ann. Math* 62 (1955) 374-410

[22] Witkin, Andrew P., "Intensity-based edge classification," *Proceedings of AAAI-82*, Pittsburgh, August 1982, 36-41.

GC970145-27

Solving the Depth Interpolation Problem with the Adaptive Chebyshev Acceleration Method on a Parallel Computer

Doug J. Choi and John R. Kender
Department of Computer Science
Columbia University
New York, N. Y. 10027

Abstract

This paper discusses solving the depth interpolation problem using the adaptive Chebyshev acceleration method on a parallel computer to speed convergence. Many low level computer vision problems, including depth interpolation, can be cast as solving a symmetric positive definite (SPD) matrix. Usually, the resulting SPD matrix is sparse. We first show the derivation of the adaptive Chebyshev acceleration method when applied to any SPD matrix. We show further how the adaptive Chebyshev acceleration method for sparse SPD matrices can be run on a particular parallel architecture (fine grained, mesh- and tree-connected, SIMD), where the Jacobi method is chosen as the underlying basic iterative method. Lastly, we show some preliminary simulation results for synthetic images, and compare them with the results from one of the commonly used methods, the Gauss-Seidel method. We also detail our future plans.

1. Depth Interpolation

Human perception is a vivid one of dense and coherent surfaces in depth. This suggests that there exists a visible-surface reconstruction process that transforms sparse information into a dense surface representation. The low level visual processes provide several visual cues to reconstruct the visible surfaces. In particular, one low level visual process, stereo, generates depth only at scattered points, another process, such as photometric stereo, generates orientation at points that may be scattered as well. With these sparse constraints, a depth interpolation process would compute the depth of the visible surfaces at every point explicitly.

Grimson formulated one approach to the depth interpolation problem [Grimson 81]. He suggested that given a set of scattered depth constraints corresponding to points along the zero-crossing contours of the primal sketch, the surface which 'best' fits the known constraints is that which passes through the known points and minimizes the expression referred to as the quadratic variation of the surface. He used a gradient descent method to find such a surface but slow convergence rates were observed in his work.

Terzopoulos worked further on surface representation [Terzopoulos 84]. The discrete form of the visible-surface reconstruction problem is described as the solution of a large sparse linear system of equations. The nonzero coefficients of each equation is specified as summations of computational molecules. Given the depth constraints and the orientation constraints, a set of computational molecules computes the nonzero coefficients of the linear system by local computations involving simple multiplications and additions of nodal variables in a specified spatial arrangement. Because of the symmetric nature of the computational molecules, it can be easily shown that the resulting matrix is symmetric. Furthermore, Terzopoulos shows the stronger result that the matrix generated is symmetric and positive definite (SPD). The matrix is also sparse. Even for nodes which are sufficiently distant from a boundary where the depth is discontinuous, they interact with only 12 neighbors, all of them at most only 2 nodes away. Terzopoulos used a multi-grid approach with the Gauss-Seidel relaxation method at each relaxation sweep to speed up the convergence rate.

In this paper, we follow the Terzopoulos' formulation on visible surface reconstruction and use the computational molecules proposed by him. However, we present an alternative depth interpolation process using the adaptive Chebyshev acceleration method, which speeds convergence and is amenable to certain classes of parallel computers. We were led to this form of acceleration by the observation of [Traub 84] that the iterative methods on one of which it is based, the Chebyshev and conjugate gradient methods, are provably optimal in terms of computational complexity.

2. Adaptive Chebyshev Acceleration Method

In the previous section, the depth interpolation problem has been cast as solving a set of linear equations,

$$Au = b \qquad (1)$$

where A is SPD.

Using one of several known basic iterative methods, it can be solved by the matrix equation

$$u^{(n+1)} = Gu^{(n)} + k, \; n = 0,1,2, \ldots \qquad (2)$$

We first review the mathematics involved in matrix iteration. The interested reader can find further details in [Young 81].

[1]This research was sponsored in part by the Defense Advanced Research Projects Agency under contract N00039-84-C-0165 and in part by an NSF Presidential Young Investigator Award.

There are several well known basic iterative methods the Jacobi method, the Gauss-Seidel method, the successive overrelaxation (SOR) method, and the symmetric successive overrelaxation (SSOR) method. However, methods other than these basic iterative methods are used in practice because of the slow convergence rates of the basic iterative methods. The rates of convergence can be accelerated by two major classes of acceleration: polynomial acceleration methods or nonpolynomial acceleration methods. Note that the multi-grid method used by Terzopoulos is one of the nonpolynomial acceleration methods.

The iterative method in (2) is *symmetrizable* if for some nonsingular matrix W the matrix $W(I - G)W^{-1}$ is SPD. If the iteration method is symmetrizable, it can be shown that the largest eigenvalue $M(G)$ of G is less than 1.0, this provides us with an useful upper bound for this critical quantity on which speed of convergence depends. Furthermore, other properties of the matrix G turn out to be sufficient for the effective use of polynomial acceleration methods, such as the Chebyshev or the conjugate gradient methods.

In general, the Jacobi method and the SSOR method are symmetrizable while the Gauss-Seidel method and the SOR method are not. However, note that it can be shown that the Gauss-Seidel method always converges when A and D are SPD, where D is a diagonal matrix whose diagonal elements are taken from the matrix A.

In our implementation, we chose the Jacobi method as the underlying basic iterative method since it is much simpler than the SSOR method, another symmetrizable basic iterative method. In particular, the sparsity of matrix is preserved, which lends itself efficiently to parallel computation, described in detail in the next section. In the Jacobi method, the matrix G is related to the matrix A by

$$G = (g_{i,j}) \quad g_{i,i} = 0 \quad \text{if } i = j \quad (3)$$

$$g_{i,j} = a_{i,j}/a_{i,i} \quad \text{if } i \neq j$$

When A is SPD, the Jacobi method is symmetrizable with $W = D^{1/2}$.

In our work, we chose the adaptive Chebyshev method as the method of polynomial acceleration. The convergence rate of this method is fastest when the largest eigenvalue, $M(G)$, and the smallest eigenvalue, $m(G)$, of the iteration matrix G for the related basic method is known.

Lee investigated using the pure Chebyshev method on several low level vision problems [Lee 85]. For shape from shading and optical flow problems, he calculated the lower and upper bounds of the smallest and largest eigenvalues. In the depth interpolation problem we investigated, we could not estimate the bounds of eigenvalues due to the flexible nature of the matrix. This is because the matrix is sensitive to the shape of the underlying region and the physical locations of known depth values. For example, at nodes where the depth constraints exist both the right and the left hand side of the matrix equation, $Au = b$, are modified. In general, the optimal estimates of the eigenvalues are not known a priori but can be determined by using the adaptive Chebyshev acceleration method.

In our work, we implemented two adaptive Chebyshev acceleration methods as given in [Young 81]. In one algorithm (Algorithm 6-4.1 on page 107), the initial estimate of $m(G)$, m_E, is input and is not changed throughout computation. The estimate of $M(G)$, M_E, is updated upward. In the other algorithm (Algorithm 6-5.1 on page 117), estimates of both eigenvalues are updated. When the initial m_E is too high, it is adjusted downward. If not, it is not changed. The other estimate, M_E, is updated in the same fashion as in the previous algorithm. As both values approach their true values, the algorithm's rate of convergence increases.

When we have a case where $m_E \leq M_E \leq M(G) < 1.0$ and $m_E \leq m(G)$, one of the ways to compute the initial estimate of $m(G)$ is by computing a reasonable lower bound based on the matrix norm, that is, $m(G) \geq -\|G\|_\infty$.

For the details of algorithms, reader is referred to [Young 81]. In the next section where we discuss the possible implementation on a parallel architecture, some computational steps of the adaptive Chebyshev acceleration method will be discussed in more detail.

3. Implementation on Parallel Architecture

Here, the proposed solution demands three characteristics of the supporting hardware. As we have shown in the previous section, we use the Jacobi method, which is symmetrizable, to convert the A matrix to the G matrix. Since we use the Jacobi method as the underlying basic iterative method, each matrix iteration should occur simultaneously for every node. The first property then is that it naturally leads to a single instruction multiple data stream (SIMD) mode of execution.

Secondly, the matrix G is sparse. In particular, as we have noted, in the depth interpolation problem even a node far removed from the region boundary interacts only with 12 neighboring nodes. Therefore, mesh interconnections between nodes are sufficient for handling all the communication needs for multiplication by G.

Thirdly, what is needed is a fast global summary capability. In the adaptive Chebyshev acceleration method we calculate various matrix and vector norms either to test for convergence or to obtain better estimates of the eigenvalues. This communication need can be met well by a tree topology, superimposed on the underlying mesh.

Two parallel architectures support these three needs well. Both NON-VON and the Connection Machine support SIMD control and mesh interconnections. In NON-VON, a tree topology is explicitly supported, see [Shaw 84]. In the Connection Machine, a binary N-Cube is superimposed on the mesh, but a tree structure can be simulated by software, such as the *calculated tree* mentioned in [Christman 84].

3.1. Parallelization

The parallelization of the adaptive Chebyshev acceleration computation is now discussed in detail. The computation proceeds in two stages: pre-computation, and iterations. At the pre-computation stage, we compute the matrix A using a set of computational molecules in SIMD fashion with four types of given inputs: depth discontinuities, depth constraints, orientation discontinuities, and orientation constraints. For each node, it computes the necessary multiplication factors for each of 12 neighboring nodes and itself. The right-hand side vector b is also computed at this time. Once the matrix A is computed, an initial estimate of $m(G)$ is also computed using the tree connections by calculating the sup-norm of G ($||G||_\infty$).

At each iteration, computation goes through several steps. Here, our attention is focused on the calculation of the next iterate. The major computations are [Young 81]:

$$\delta = Gu_{cur} + k - u_{cur};$$

$$\text{DELTP} = ||\delta||_2; \quad \text{DELTE} = ||\delta||_\beta;$$

$$u_{nxt} = \rho(\gamma\delta + u_{cur}) + (1.0 - \rho)u_{prv};$$

$$\text{TUB} = ||u_{nxt}||_\eta;$$

$$u_{prv} = u_{cur}; \quad u_{cur} = u_{nxt};$$

Computation of u_{prv}, u_{cur}, and u_{nxt} are straightforward. Each node stores each of these numbers so that a simple SIMD execution will update each one, independent of other nodes. No communication is needed. The values ρ and γ are iteration parameters of the Chebyshev method. They are determined by the current estimates of the eigenvalues, m_E and M_E.

Calculation of the 2-norm of e, the β-norm of δ or the η-norm of u_{nxt} are handled well using a tree topology. Usually, the β-norm or η-norm is either a 2-norm or sup-norm. When the 2-norm is needed, every element in each PE is multiplied with itself and the summation of squared numbers is carried out from the bottom to the top of the tree, one level at a time. The square root of the final resulting sum obtained at the top is the value desired. (When the size of the mesh at the bottom of the tree is $n \times n$, the entire process takes $\log n$ steps.) When the sup-norm is desired, each PE calculates the absolute value of the element in it, and then compares its own value against those values of its own two sons. Both the comparison and the retaining of the biggest value is similarly carried out from the bottom to the top of the tree. The single value obtained at the top is the sup-norm desired. (This too is a $\log n$ process.)

Finally, we're left with the calculation that involves the matrix multiplication operation. Computation of the pseudoresidual vector δ can be done in SIMD fashion using mesh interconnections only. The only step remaining at this point is the computation of matrix multiplication term, Gu_{cur}. In the previous section, we have already seen that the Jacobi method is parallel (i.e., it simultaneously displaces old values with new values). Therefore, iterations based on the Jacobi method can be carried out in SIMD fashion with mesh interconnections to assemble current depth values of neighboring nodes. Furthermore, all the coefficients that contribute to this assembly are in the form

$$g_{i,j} = a_{i,j}/a_{i,i}$$

since i is not equal to j. Since the factor in denominator $a_{i,i}$ is common to all neighboring nodes, division by it is done only once, as the last step. By using the Jacobi method, neither explicit pre-computation of the matrix G nor any particular sophisticated ordering of matrix elements is needed. Put in other words, only local computation supported by mesh topology is all that needed to carry out iterations when vector u_{cur} is multiplied by the matrix G.

4. Simulation Results

We have extended the existing NON-VON simulator [Hussein 85] to handle floating-point operations. It was straightforward to implement and test the SIMD control, mesh connections, and tree topology aspects of the computation. We later simulated the Gauss-Seidel method, which requires only mesh connections, for comparison.

In our preliminary simulation work, the synthetic image was a constant depth plane ($u = 1.0$). The shape of the boundary of the plane was a square, with size 10 x 10. The depth constraints were scattered randomly over the plane. The density of the depth constraints were 15%, 30%, and 50%.

For the bulk of our simulation work, we used the first Young algorithm (Algorithm 6-4.1 in [Young 81]). In this algorithm, the estimate of $M(G)$ is updated upward during iterations but the estimate of $m(G)$ is not changed at all. When the estimates are not optimal, the convergence rate is not fast at the beginning. When the estimate M_E is close enough to $M(G)$, the convergence is fast even in the case that m_E is not close enough to $m(G)$; in the adaptive Chebyshev acceleration method, M_E is more critical than m_E. Therefore, before embarking on this algorithm, we carried out an experiment to see how much we lose by running this algorithm with the reasonable and simple to compute initial estimates of $M_E = 0$ and $m_E = -||G||_\infty$ [Young 81]. In the experiment, we used a 10 x 10 image where 50% of the nodes were constrained. For the initial value of m_E, the calculated lower bound was -3.0. In this case, the algorithm took 287 steps to converge. In contrast, we ran the algorithm a second time, using the best problem-specific estimates obtainable, -0.4 for the initial value of m_E and .9975 for the initial value of M_E. The run with better initial estimates took 145 steps, i.e., only half of time, to converge. This result should be taken as a rather conservative one. With bigger images, the difference of the number of steps to convergence should become smaller, i.e., the adaptive Chebyshev acceleration method arrives at optimal convergence rate relatively quickly.

(A note on eigenvalues. To get these best estimates, we used the other algorithm of Young (Algorithm 6-5.1 in [Young 81]). In this algorithm, the estimate of $M(G)$ is updated upward while the estimate of $m(G)$ is updated downward if current estimate, $m(G)$, is bigger than the smallest eigenvalue, $m(G)$. To obtain the best estimate of $M(G)$, we picked a number that was slightly smaller than the final estimate at convergence, to obtain the best estimate of $m(G)$, we picked a large enough number, say, -1. Then we ran the

algorithm repeatedly and checked whether the current estimate was updated downward while the algorithm was running. We continued this until both estimates were not updated at all throughout the entire computation. We also verified that the adaptive Chebyshev acceleration method took the smallest number of steps to converge when these best estimates were input as the initial values.)

To compare our approach with existing methods, we also ran the same image (10 x 10 with 50% depth-constraint density) with the Gauss-Seidel method. In general, the depth values approximate their final positions more rapidly in the beginning for the Gauss-Seidel method but eventually the adaptive Chebyshev acceleration method catches up. Thereafter, the depth values are closer to ideal values for the adaptive Chebyshev acceleration method. In other words, eventually the Gauss-Seidel method converges more slowly. It took 304 iterations for the Gauss-Seidel method to converge. Recall that it took 267 iterations for the adaptive Chebyshev acceleration method to converge, while the adaptive Chebyshev acceleration method with the best estimates took 145 iterations to converge.

4.1. Convergence Properties

To measure convergence, the convergence criterion for the adaptive Chebyshev acceleration method was applied to the Gauss-Seidel method as well. The iteration error vector is obtained in the abstract by subtracting the ideal depth vector from the current depth vector computed. The iterations are to be terminated whenever some norm of the error vector becomes sufficiently small. In the adaptive Chebyshev acceleration method, the norm of the error vector is proportional to the norm of the pseudo-residual vector, provided that the current estimate, $M(E)$, is close enough to $M(G)$. Often, a relative error measure is desired rather than an absolute error measure. In this case, the β-norm of the pseudo-residual vector divided by the α-norm of the depth vector is compared to stopping criterion number instead of the β-norm of the pseudo-residual vector alone. For more detailed discussion of the convergence criterion of the adaptive Chebyshev acceleration method, the reader is referred to [Young 81]. Since we extended the convergence criterion of the adaptive Chebyshev acceleration method to the Gauss-Seidel method, the number of iterations may not be the best measure for comparison.

Thus, we looked into another measure of comparative performance. When the Gauss-Seidel method was stopped after the same number of iterations as the adaptive Chebyshev acceleration method took to converge, it was observed that the final depth values obtained from the adaptive Chebyshev acceleration method were closer to ideal depth values of synthetic image for most of nodes. Returning to the experiment we mentioned before, (10 x 10 image with 50% depth-constraint density) for 84 nodes out of 100, the adaptive Chebyshev acceleration method produced better depth values.

However, an anomaly of convergence was observed as well. For the 50% density case, the *average* depth value was slightly further from the ideal depth value for the adaptive Chebyshev acceleration method. It was due to the fact that some remaining nodes did not converge well for the adaptive Chebyshev acceleration method. This phenomena was more apparent when we examined the minimum and the maximum final depth values. When the density got lower, it got worse. For the Gauss-Seidel method, the convergence rate was rather slow but the final depth values obtained were much more uniform. In close inspection, it was revealed that those ill-behaving nodes were clustered along a row or a column where depth constraints did not exist along the line.

The numerical values are summarized in Table 1. There, useful measures are listed for three different densities of depth constraints for both the adaptive Chebyshev acceleration and the Gauss-Seidel method. They are the number of iterations, the sup-norm of the pseudo-residual vector, the minimum, the maximum, and the average of the final depth values, and the final estimate M_E (for the adaptive Chebyshev acceleration method only).

5. Conclusion and Future Plans

In this paper, we showed how the adaptive Chebyshev acceleration method can be applied to parallel computers for those computer vision problems where the resulting matrix is SPD. Basically, the speed-ups of matrix iteration have been achieved by two factors. First, we used a theoretically better method, i.e., the adaptive Chebyshev acceleration method. Second, all computational steps have been parallelized.

In this paper, a synthetic image with constant depth was described. Other synthetic images will be pursued in future. Our implementation examples were quite small compared to real images. Currently, our simulator runs on both DEC-20 and VAX 11/750. By transporting our software to a bigger machine, we will be able to handle bigger images.

We have compared our simulation results with the Gauss-Seidel method, our results will also be compared with the results obtained from the multi-grid method. We also intend to solve these problems using the pure iteration methods on which the adaptive Chebyshev acceleration method is based, namely pure Chebyshev method, and the conjugate gradient method. Thus, we will compare these approaches, some of which are more suited to particular architectures.

Acknowledgements

D. E. Shaw's laboratory has provided us with the simulator and the prototype machines, which has made this research possible. G. W. Wasilkowski provided valuable comments on the mathematics.

References

[Christman 84] Christman, D. P., *Programming the Connection Machine*, Master's thesis, Massachusetts Institute of Technology, January 1984.

[Grimson 81] Grimson, W. E. L., *From Images to Surfaces*, MIT Press, 1981.

[Hussein 85] Hussein, A. H. I., *Image Understanding Algorithms on Fine-Grained Tree-Structured SIMD Machines*, Ph. D Thesis, Columbia University, 1984.

[Lee 85] Lee, D., *Contributions to Information-based Complexity, Image Understanding and Logic Circuit Design*, Ph. D. Thesis, Columbia University, 1985.

[Shaw 84] Shaw, D. E., *Organization and Operation of a Massively Parallel Machine*, Tech report, Columbia University, 1984.

[Terzopoulos 84] Terzopoulos, D., *Multiresolution Computation of Visible-surface Representations*, Ph. D Thesis, Massachusetts Institute of Technology, January 1984.

[Traub 84] Traub, J. F., and Wozniakowski, H., *On the Optimal Solution of Large Linear Systems*, Journal of ACM 31(3) 545-559, 1984.

[Young 81] Young, D. M., and Hageman, L. A., *Applied Iterative Methods*, Academic Press, 1981.

Results from the adaptive Chebyshev acceleration method

density	steps	$\|\delta\|_\infty$	u_{min}	u_{max}	u_{avg}	M_E
15%	1041	1.4920E-6	-.32020117	1.3772003	.65242547	.999879
30%	1171	7.9166E-7	-.48583681	1.0003357	.92526618	.999917
50%	287	1.4965E-5	.99025001	1.0003906	.99960784	.998399

Results from the Gauss-Seidel method

density	steps	$\|\delta\|_\infty$	u_{min}	u_{max}	u_{avg}
15%	1685	1.0610E-6	.99997884	1.0000081	.99999444
30%	981	7.8846E-7	.99998441	1.0000112	.99999496
50%	304	1.5004E-5	.99965197	1.0001929	.99987541

Table 1: Simulation Results for the 10 x 10 image

CMU-RI-TR-86-5

First Results on Outdoor Scene Analysis Using Range Data

Martial Hebert
Robotics Institute

and

Takeo Kanade
Computer Science Department

Carnegie Mellon University
Pittsburgh PA15213

Abstract

This paper describes some techniques for outdoor scene analysis using range data. The purpose of these techniques is to build a 3-D representation of the environment of a mobile robot equipped with a range sensor. Algorithms are presented for scene segmentation, object detection, and map building.

We present results obtained in an outdoor navigation environment in which a laser range finder is mounted on a vehicle. These results have been applied to the problem of path planning through obstacles.

1. Introduction

This paper presents some techniques developed for outdoor scene analysis using range data. The advantages of using such range data fall in two categories: First, range data is less sensitive to environmental conditions, such as lighting, thus alleviating shadow or highlight problems. Second, range data directly provides information about the geometry of the scene, such as the position of a particular feature, thus suppressing the calibration and registration steps required with camera data. This property is especially important in the area of mobile robots in which the output of the vision programs must be converted into usable space coordinates. For our work, we use the ERIM laser range sensor which provides reasonable accuracy, a working space large enough for outdoor applications, and high acquisition speed.

This research is supported by the Defense Advanced Research Projects Agency (DOD) and monitored by the Office of Naval Research under Contract N00014-82-K-0193 and by contract DACA76-85-C-0003 issued by the U. S. Army Engineer Topographic Laboratories.

Following the description of the ERIM range sensor, we present preprocessing techniques for removing sensor-dependent defaults in section 3. In section 4, we present the 3-D features extraction algorithms which are designed to produce relevant features for outdoor vehicle navigation and object recognition. 3-D map reconstruction from range data is described in section 5.

2. Sensor Description

The range sensor we use for outdoor scene analysis has been designed by the Environmental Research Institute of Michigan and will be referred to as the ERIM sensor. The basic principle of the sensor is to determine the range from the sensor to the scene point for each pixel by measuring the transmit time of a modulated laser beam. The transmit time is derived by measuring the phase difference between the reference and reflected signals which corresponds to the range from the sensor to the target. A two mirrors scanning mechanism directs the beam onto the scene so that an image of the scene is produced. In the ERIM ALV version, the field of view is $\pm 40^{\circ}$ in the horizontal plane and 30° in the vertical plane, from 15° to 45°. The resulting range image is a 64 × 256 8 bit image. The frame rate is currently two images per second (1/2 second/frame). The nominal range noise is 0.1 feet at 10 feet.

Since only the phase shift is measured, the resulting values are relative instead of absolute measurements. That is, two points separated by a length equal to a complete phase shift have the same range value. This critical length is called the ambiguity interval and is equal to 64 feet.

The sensor is also capable of producing reflectance images in which the value of each pixel is the amount of light reflected by the target. This information has not been used yet.

Figure 2-1 shows a sequence of eight ERIM images taken in a park: two consecutive images are taken from positions separated by approximately five meters.

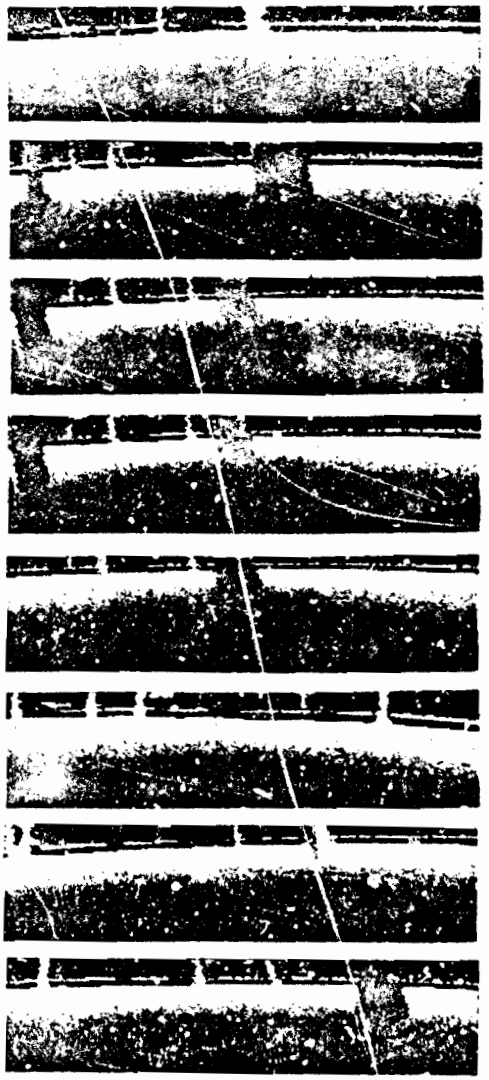


Figure 2-1: A sequence of ERIM images

3. Preprocessing

The ERIM data includes a periodicity problem due to the ambiguity interval. The periodicity is especially apparent in images such as the one shown in figure 3-1 in which distant points have the same value as close points. This problem reduces the range at which the scene can be processed to the extent of the ambiguity interval, and may also create false features, such as false edges which do not correspond to any physical feature. Therefore, the first step in the ERIM image processing is to remove the periodicity. The periodicity removal algorithm has three steps:

- Divide the image into connected components so that two points whose range difference is greater than a threshold are never connected (figure 3-2). Two such points belong to two different ambiguity intervals.
- Remove the small regions which correspond to noise.
- Explore the graph of components starting at the bottom region which is within the first ambiguity interval. During the exploration, add an offset to all the points of the currently visited region. Initially, the offset is zero, then it is incremented by 256 each time a region above the current region is visited. The result of the correction is shown in figure 3-3.

We have found that this algorithm works well within the two first

Figure 3-1: Non Corrected image

Figure 3-2: Ambiguity intervals

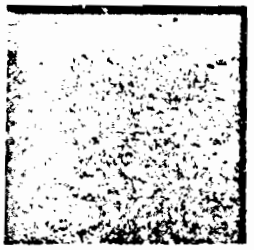


Figure 3-3: Corrected image

ambiguity intervals. Beyond that point, measurements are usually too noisy to ensure reliable results. No algorithm is guaranteed to retrieve the actual range values since it is unknown whether two regions are separated by only one or several ambiguity intervals. The algorithm assumes that only one interval separates two regions.

4. 3-D Features Extraction and Image Interpretation

In this section, we describe the feature extraction techniques. The basic features relevant to the outdoor navigation problem are:

- *3-D edges* which correspond either to discontinuities of depth such as the boundary of an obstacle, or to discontinuities of surface normalssuch as the boundary between a flat region and a highly curved region.
- *Smooth regions* which have a low curvature. These regions are further divided into accessible regions and obstacle regions.
- *Obstacles* which are either high curvature, ground regions, or objects divided into smooth or pseudo-planar regions.

The output structure from this feature extraction process is a connectivity graph of these features. The feature extraction proceeds by first computing low-level attributes (edges, normals, and curvatures) from the image, and then merging the segmentations derived from these attributes into a single description (figure 4-1).

4.1. Surface Normals

The surface normals provide important pieces of information about the shape of the observed terrain. The best way of computing the surface normals is to approximate the neighborhood of each pixel by a plane. A straightforward method would be to minimize for each pixel:

$$\sum_{i,j \in N} \alpha_{ij} \| \vec{\nu} \vec{p}_{ij} - D \|^2 \qquad (1)$$

where N is the size of the neighborhood, $\vec{\nu}$ is the surface unit normal, D is the normal distance between the origin and the plane, α_{ij} are weighting factors, and $\vec{p}_{ij}$ are the measured points. Although simple, this procedure is time-consuming. Moreover, it does not take into account the fact that the ERIM scanner delivers the radial distances instead of the Cartesian coordinates. The alternative and preferred criterion is:

$$\sum_{i,j \in N} \alpha_{ij} \| \vec{\mathcal{V}} \cdot \vec{u}_{ij} - \frac{1}{d_{ij}} \|^2 \qquad (2)$$

where $\vec{\mathcal{V}} = \vec{\nu} / D$, $\vec{u}_{ij}$ is the radial vector at pixel (i,j), and d_{ij} is the distance from pixel (i,j) to the origin.

The solution of (2) is given by:

$$\vec{\mathcal{V}} = M^{-1} \sum_{i,j \in N} \alpha_{ij} \frac{\vec{u}_{ij}}{d_{ij}} \qquad (3)$$

Where M is the 3×3 matrix defined by

$$M = \sum \alpha_{ij} \vec{u}_{ij} \vec{u}_{ij}^{t}$$

Since the vectors $\vec{u}_{ij}$ depend only on the scanning parameters of the sensor, the matrix M^{-1} can be computed *beforehand*.

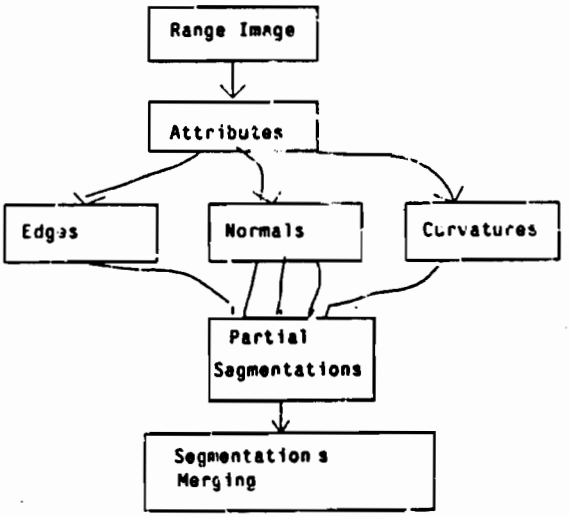


Figure 4-1: Feature extraction process

Actually, the vectors and matrices depend also on the orientation of the sensor (pan and tilt angles), but their value can be updated easily: if the sensor is rotated by a rotation R, then the radial vectors $\vec{u}$ and the resulting normal $\vec{\mathcal{V}}$ are changed to $R^t \vec{u}$ and $R^t \vec{\mathcal{V}}$, respectively.

In summary, the estimation of the surface normals proceeds as follows:

- If the orientation of the sensor has been changed since the last image, update the vectors $\vec{u}$'s.
- Correlate the inverse image $1/d_{ij}$ with $\vec{u}$ with weights α_{ij}.
- Multiply the resulting vector image by M^{-1}.
- Normalize the resulting vector to obtain the unit surface normal.

Figure 4-2 shows a range image and the three components of the surface normal estimated by using a 5 x 5 window.

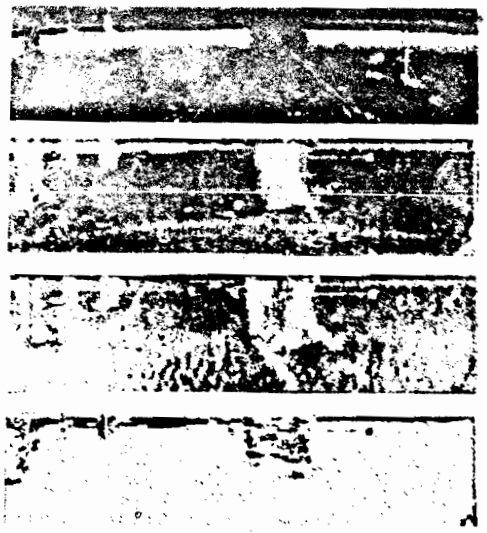


Figure 4-2: Original range image and surface normals

4.2. Curvatures

4.2.1 Using Principal Curvatures

Several authors have shown that differential geometry, namely the theory of principal curvatures, can be used to recover properties of a surface observed for a range sensor [6, 5]. These properties range from the extraction of roof edges to the extraction of cylindrical surfaces. However, these techniques do not work well for outdoor imagery. the perequisites of these techniques are that an accurate estimation of second order differential attributes is possible and that the surfaces are mathematically well defined. While outdoor imagery has a limited accuracy, and the observed surfaces usually do not have a well-defined mathematical representation since in a natural environment, most surfaces, such as a grassy terrain and tree foliage, are highly textured and irregular. Therefore, we limit ourselves to the computation of curvatures for the purpose of roughly segmenting the scene into separate regions, each of which is a region of low and uniform maximum curvature.

4.2.2. Computing Curvatures

The computation of principal curvatures can be reduced to the computation of first and second derivatives of the image (see [2] for a complete definition). The standard way of computing the curvature is to define the range image as a function $Z = f(X, Y)$, where the two coordinates X and Y are assumed to be uniformly distributed along the image rows and columns. This assumption is not true when using the ERIM sensor because of the width of the field of view. The solution is to use the spherical representation $\vec{p} = d\vec{u}(\varphi,\theta)$, where $\vec{p}$ is a measured point on the surface, d is the range given by the ERIM sensor, and $\vec{u}(\varphi,\theta)$ is the radial vector of angles φ and θ. A surface is considered as a parametric surface $\vec{p} = f(\varphi,\theta)$. The curvatures can be computed by using the derivatives with respect to the two angles:

$$\frac{\partial f}{\partial \varphi} = \frac{\partial d}{\partial \varphi}\vec{u}(\varphi,\theta) + d\frac{\partial \vec{u}(\varphi,\theta)}{\partial \varphi} \qquad (4)$$

$$\frac{\partial f}{\partial \theta} = \frac{\partial d}{\partial \theta}\vec{u}(\varphi,\theta) + d\frac{\partial \vec{u}(\varphi,\theta)}{\partial \theta}$$

.....etc

The radial vectors $\vec{u}(\varphi,\theta)$ depend only on the characteristics of the sensor and its orientation. These vectors can therefore be computed *beforehand*. The curvatures computation is thus reduced to the computation of the first and second derivatives of the image d with respect to the angles φ and θ. This computation is done by first applying a Gaussian smoothing, and then computing the derivatives by convolving with 3 x 3 masks. These masks are derived from a second order approximation of $\vec{p} = f(\varphi,\theta)$. In summary, the curvature computation algorithm proceeds as follows:

- Apply a Gaussian smoothing on the range image.
- Compute the derivatives of the range image with respect to the two spherical angles by applying 3 x 3 operators.
- Derive the derivatives of the three Cartesian coordinates by using equations (4).
- Derive the curvatures by applying the fundamental forms equations [2].

Figure 4-3 shows a range image and the corresponding maximum curvature image estimating by using the above algorithm.

4.3. Edges

Edges are computed by detecting the depth jumps. This is done by first detecting the zero-crossings of a difference of Gaussian masks in the depth image. One difficulty is that if 2 pixels are far from the sensor, they may appear as neighbors in the image even though they may be far apart in space and hence do not correspond to edges. This problem is overcome by considering only points of the image with sufficiently high

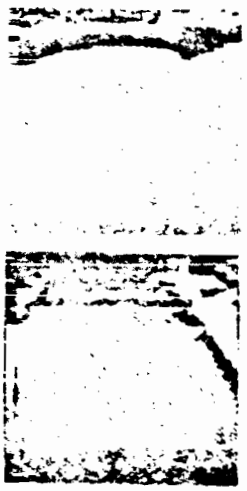


Figure 4-3: Original range image and curvature image

curvature as potential edge points. This algorithm detects mainly the jump edges corresponding to the boundaries of vertical obstacles. A more elaborate edge finder for detecting discontinuities of the surface normals is based on region segmentation.

4.4. Segmentation Algorithm

The segmentation algorithm proceeds by combining partial segmentations obtained from attributes, such as edges, surface normals and curvatures, into a consistent scene segmentation (figure 4-1). The advantage of this approach is that it takes into account all the available information while dividing the whole segmentation problem into smaller ones.

The initial segmentation are produced for each attribute. 3-components surface normals and curvatures by a three steps region growing algorithm:

1. Find clusters in the attribute space such as clusters in the surface normals space.
2. Identify the regions corresponding to those clusters in the original image.
3. Use these regions as starting regions for a region growing algorithm. The edges are used as region boundaries in the region growing.

The segmentations obtained from individual attributes are then merged together. This merging step compares each region of one segmentation to the corresponding region in the image of the other segmentations: if the segmentations agree, the region is reported, otherwise it is split into connected regions consistent with the other segmentations. Figure 4-4 shows the segmentation of a short sequence of ERIM images. The regions are labeled as *smooth*, i.e., accessible portions of the terrain, and *obstacle*, i.e., objects in the scene. The first principal direction of a region is attached to the region in the display.

5. Surface Reconstruction and 3-D Map Building

In the previous section we presented techniques for extracting information for ERIM images. In this section, we present techniques for storing this information as a local 3-D *map* of the environment. A map is a structure describing the geometry of the environment from which one can derive information such as the type of terrain at a given space location *x,y,z*.

5.1. Local Map

One important characteristic of a range image is that it permits the production of a three dimensional map of the current local environment. Such a map, called a local map, is derived from only one image and can be viewed as the local state of the environment. This map can in turn be used to predict the appearance of the scene from another viewpoint, and to plan a safe path for a vehicle while taking into account the 3-D shape of the traversed terrain. The path planning is especially important in cross-country navigation where the "flat ground" assumption does not usually hold.

Figure 5-1 shows an example of such a map and the corresponding ERIM image. The map is displayed as a 3-D mesh of measured points with the surface normal at each point.

5.2. Global Map

All the techniques described so far proceed by independently processing one image at a time. In an outdoor navigation system, consecutive images are related to each other to develop a global map. That is, the robot grabs an image every 1 to 10 meters as in the sequence shown in figure 2-1. Then each image is registered with respect to the previous ones. In other words, we merge the local maps produced by each individual image into a global map describing the environment explored so far. Such a global map can be used for two main purposes:

- *Incremental description refinement.* A single image provides only partial information about the identified objects. For example only the front part of the tree is visible in figure 5-1, and, as a result, no information is available in the cone shaped unknown region behind it. Putting together several local maps obtained from several different viewpoints would refine the objects description by reducing the size of the unknown regions.

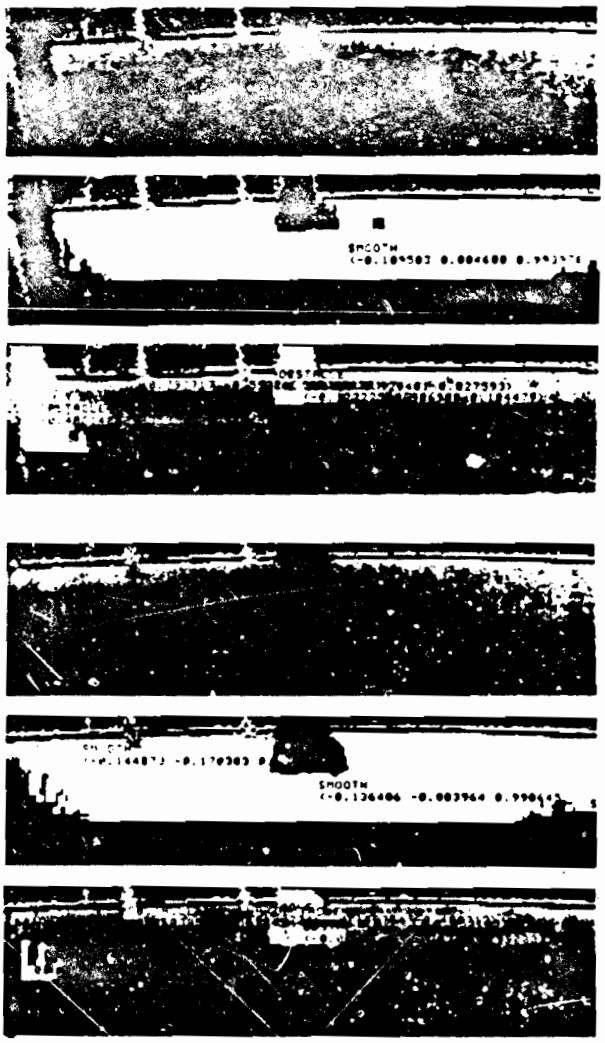


Figure 4-4: Segmented sequence

- *Incremental construction of a reference map.* One application of outdoor vision is the exploration scenario. A vehicle equipped with sensors discovers an unknown environment and stores information in a reference map usable during later missions.

Global map building is considered as a matching process between consecutive images:

1. Assume that frame 1 is part of the global map and is described in some global coordinate system.
2. Extract features from frame 2. These features are linear regions and edges approximated by polygonal chains.

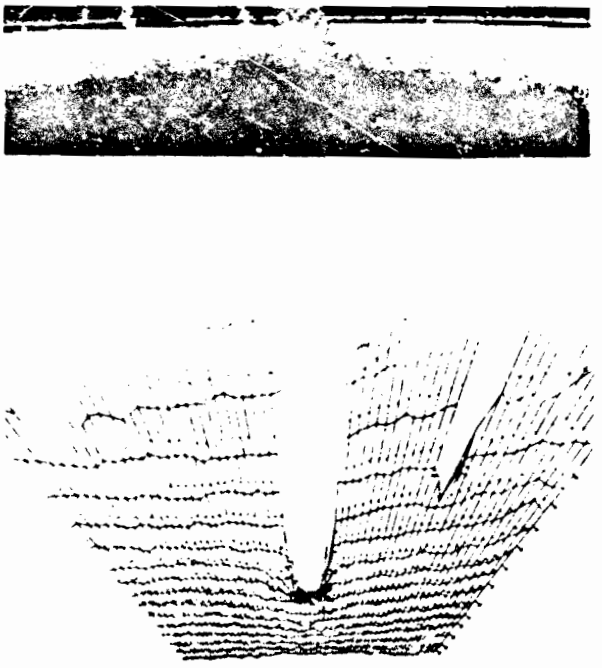


Figure 5-1: Local map derived from an ERIM image

3. Match the features of images 1 and 2 by using a tree search procedure guided by the transformation estimation. This procedure is similar to the ones described in [1] and [3].
4. The matching provides an estimate of the transformation between images 1 and 2, which in turn provides an estimate of the position of image 2 with respect to the global map.
5. Include the local map derived from image 2 into the global map. That involves the identification of overlapping regions and the updating of objects descriptions.

We have done experiments on steps 1 through 4 using ERIM data. Figures 5-2 and 5-3 show two consecutive frames. The matching of the two images leads to a transformation estimate which is applied to the second local map. Figure 5-4 shows the two registered local maps. Finally, overlapping regions are identified, thus leading to the updated map of figure 5-5. In this example, the map updating has been local at the pixel level and does not include the updating of the symbolic description.

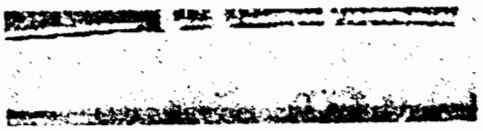


Figure 5-2: First image

Figure 5-3: Second image

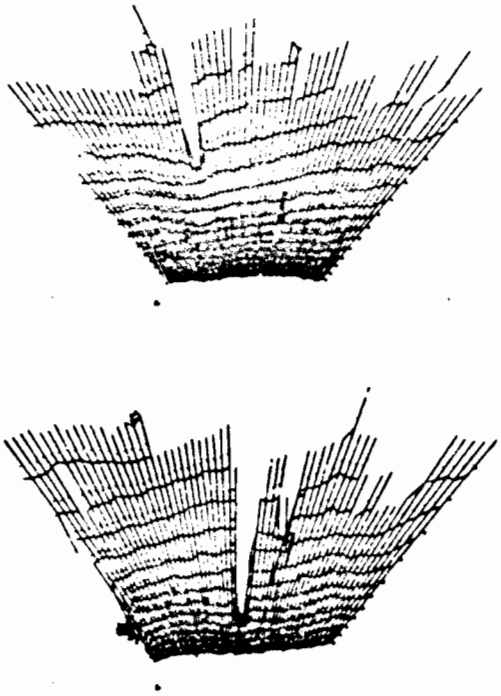


Figure 5-4: Registered local maps

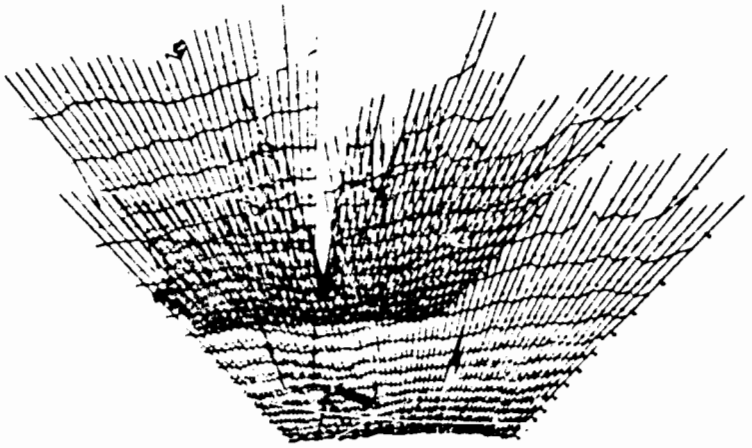


Figure 5-5: Global map

Matching the features is efficient since the number of features is usually small, and the transformation between images is partially known beforehand. That is, bounds on the dislocement between two consecutive frames are available.

6. Conclusion

The techniques presented in this paper have been experimented in a realistic outdoor environment by mounting the sensor on a mobile robot [4]. The results indicate that active range data processing is suitable for the navigation through an unknown environment. Future work includes the combination of range data with other sources of visual data, such as reflectance or color, and the 3-D object recognition, that is, the identification of specific objects in the scene by matching extracted features and stored models. The combination of these techniques will provide a powerful system for outdoor scene interpretation.

Acknowledgments

Mike Blackwell installed the hardware and software environment of the ERIM sensor. The CMU Civil Engineering Lab provided the testbed vehicle. Takayoshi Obatake implemented part of the 3-D map builder. The authors wish to acknowledge the CMU Image Understanding group for helpful discussions and support.

References

[1] Faugeras O.D., Hebert, M.
A 3D recognition and positioning algorithm using geometrical constraints between primitive surfaces.
In *Proc. Eigth Int. Joint Conf. On Artificial Intelligence*, pages 996-1002. Karlsruhe, August, 1983.

[2] Faux, I.D., Pratt, M.J.
Computational Geometry for Design and Manufacture.
Ellis Horwood, 1979.

[3] Hebert, M., Kanade, T.
The 3-D Profile Method for Object Recognition.
In *Proc. CVPR'85*, pages 458-454. San Francisco, June, 1985.

[4] Kanade, T., Thorpe, C.
CMU ALV *Project Report: 1984 to 1985.*
Technical Report Forthcoming Technical Report, Carnegie Mellon University, 1986.

[5] Medioni, G., Nevatia, R.
Description of 3-D Surfaces Using Curvature Properties.
In *Proc. Image Understanding Workshop.* DARPA, Science Applications, McLean, Va., 1984.

[6] Ponce, J.,Brady M.
Toward a Surface Primal Sketch.
In *Proc. Intern. Conf. on Robotics and Automation.* St Louis, March, 1985.

GC790145-29

DESCRIPTION OF SURFACES FROM RANGE DATA[1]

T.J. Fan, G. Medioni and R. Nevatia

Intelligent Systems Group
Departments of Electrical Engineering
and Computer Science
University of Southern California
Los Angeles, California 90089-0273

ABSTRACT

Curvature properties, if known at every point on the surface, completely specify the surface. We argue that extracting significant curvature *changes* leads to a rich and robust surface representation. Since curvature computation is a highly noise sensitive operation, we smooth the image with masks of increasing size, detecting features at the smoothest level and localizing them at the original one. Our representation consists of a list of labels that are computed from the curvature properties but that correspond to significant physical properties of the surface such as jump, folds, and extrema. We illustrate our technique with several examples.

1. INTRODUCTION

We are interested in the description of 3-D surfaces and objects, assuming that range data (i.e. the 3-D positions) of the points on the visible surface are available, say by the use of a laser range finder. We also assume that this data is "dense", in the sense of being sampled on a certain grid and not just at discontinuities (as may be the case for uninterpolated edge-based stereo data).

To generate useful descriptions, we need a useful representation. In general, such a description should be suitable for the task of object recognition and position identification. It should be rich, so that similar objects can be identified, stable, so that local changes do not radically alter the descriptions, and have local support so that partially visible objects can be identified. It should also enable us to recreate, from its features, a shape reasonably close to the original one.

Generalized cones have come to be recognized as an important class of representations, that satisfy the above requirements, particularly for complex objects which are described as assemblies of smaller objects [1, 2]. However, generalized cones are "volume" descriptions and may not be suited for objects that are essentially surfaces, such as a metal sheet, or for relative smooth, "featureless" surfaces such as a turbine blade. In this paper, our interest is primarily in the description of surfaces, though we think that such descriptions will also be an important tool in generating generalized cone descriptions.

[1]This research was supported by the Defense Advanced Research Projects Agency under contract number F33615-84-K-1404, monitored by the Air Force Wright Aeronautical Laboratories. Darpa Order No. 3119

It is well known in differential geometry that the information given by the magnitude and the orientation of the principal curvatures determines a surface completely and uniquely. Our interest in this work is to use this data to isolate important physical properties of a surface.

In particular, we are interested in the following surface properties:

1. jump boundaries where the surface undergoes a discontinuity
2. folds which correspond to surface orientation discontinuities
3. ridge lines which correspond to smooth local extrema of curvature

We will show that these surface properties can be inferred from "zero-crossings" and extremal values of surface curvature properties.

Curvature properties have been used to describe curves in the past by several authors, e.g. see [3, 4]. Interest in describing surfaces in 3-D is more recent, perhaps because of unavailability of good range data in the distant past. Many authors have concentrated on pointwise descriptions of surface properties rather than extracting characteristic features [5, 6]. Our work is similar to that of Ponce and Brady [7] but differs in detail. This paper is an update of the approach we presented earlier [8] and describes new implementations and results.

The next section presents a review of previous approaches to surface representation, section 3 presents our approach in detail, section 4 shows results for various images and section 5 outlines future research directions.

2. ALTERNATIVE APPROACHES TO SURFACE DESCRIPTION

A recent survey paper by Jain and Besl [9] constitutes an excellent overview of the field of range image analysis. We can characterize previous approaches to 3-D surface descriptions in the following classes:

- approximation by "simple" surfaces, such as planar patches

- extraction of "edges" in range data
- surface characterization

2.1. Simple surfaces

Surface representation has been important in computer graphics almost from the start. The earliest approaches used approximation by planar patches (typically triangular); later methods have used other surface patches such as "Coons patches" and 2-D spline fits. We believe that such methods, while adequate for graphics applications, are rather poor for computer vision, as the approximating surfaces are subject to large changes even if the observed surfaces change only slightly. The number of patches found is typically very large and the points and lines where the approximating patches are joined need not have any physical significance.

Some examples of using surface approximations for computer vision applications can be found in [10], [11], [12], [13], [14] and [4].

2.2. Edges

Another approach is to find "edges" in surfaces directly. The jump boundaries can be easily located by specialized operators used for intensity edge detection. The more difficult problem is the detection of smooth edges such as at folds. Successful methods have been developed for detecting these features when they can be modeled as the intersection of two planar patches [15, 16], or to hypothesize the presence of known objects in the scene [17]. These features are then used in conjunction with others to generate a useful description.

2.3. 3-D surface characterization

The methods to characterize surfaces can be viewed as being in two sub-classes. The first describes the surfaces by point-wise properties whereas the others attempt to derive global descriptions. Several previous authors have described methods of characterizing surfaces by using curvature but not all were aiming for global descriptions.

Lin and Perry [18] argue that integrals of the scalar curvature of a surface provide good shape information. The computation of these quantities is done using surface triangulation. No experimental results are given in their paper.

Sethi and Jayaramamurthy [6] first compute the surface normal at each point, then extract characteristic contours, which are sets of points for which surface normals are at a constraint inclination with a reference vector. Underlying surfaces are then identified by taking a Hough transform of these contours. This approach works well for simple surface (cylinders, cones).

Laffey, Haralick and Watson [19] fit at each pixel a two dimensional cubic polynomial to estimate the first and second partial derivatives at that point. They then compute features to classify each pixel into classes such as peaks or ridges. The output is still a dense representation on which feature extraction has to be performed.

Langridge [20] gives the result of a preliminary investigation into the problem of detecting and locating discontinuities in the first derivatives of surfaces. Results are shown only on 2 simple synthetic examples.

Neckman [21] proposes to extract peaks (local maxima), pits (local minima) and passes (saddle points) of a surface, to connect these features and obtain a graph which appears useful to describe variations of a smooth surface.

Finally, Ponce and Brady [7, 4] adopt an approach very similar to the one reported here as they first smooth the surface with Gaussian masks with increasing variance σ, then compute extrema of the principal curvatures, and finally group sets of features to recognize models such as roof and step surface discontinuities. These operations are very similar to what we describe; the differences being in details of implementation (unfortunately, Ponce and Brady's paper leaves out many of the details, making a direct comparison difficult).

2.4. Our approach to using curvature properties

Our approach consists of examining properties of surface curvature to infer significant physical properties of the surface. The curvature properties that we consider are the extrema and the "zero-crossings" of the surface curvature. As surface curvature is dependent on the direction of measurement we need to perform this computation either in "all" directions, or choose specific orientations, such as those of the principal curvatures. In this paper, we do the former, in earlier work [8] we used the maximum principal curvature, which causes some difficulties as explained in section 3.1.

It can be shown that these curvature properties correspond to certain significant physical properties of a surface [7, 8]. An occluding boundary (sometimes referred to as a "jump boundary" in range data analysis) creates a zero-crossing of the curvature in a direction normal to that of the boundary. A "fold boundary" (where surface normals are discontinuous) causes a local extremum of the curvature at that point. Fold boundaries may also create zero-crossings away from the location of the boundary itself. Lastly, curvature extrema correspond to certain distinguished points or lines on smooth surfaces, such as along the extremes of the major axis of the cross-sections of an elliptical cylinder.

To turn these observations into functioning algorithms, we need to solve the following:

1. Compute curvature properties

   Since we work with digital data, we can only compute approximations of curvature, either by differencing or by fitting analytical surfaces to given data.

2. Extract and localize significant features of these curvature properties:

   Curvature, being a local measure, is highly noise sensitive. Larger masks to compute difference may be used, but they result in a loss of accuracy of localization. We follow a scale-space filtering approach [22] to help solve these problems.

3. Interpret sets of curvature features in terms of physical properties of surfaces:

   We wish to detect occluding boundaries, surface discontinuities and smooth local extrema.

While these steps are rather straightforward in concept, their implementation requires resolution of many detailed issues such as how to compute curvature, which curvature properties to use, and how to combine information from different scales. We next describe an earlier implementation [8] and its deficiencies and then a new implementation designed to overcome these problems.

3. Description of the method

3.1. Previous implementation

In our previous implementation [8], we worked primarily with the maximum principal curvature, say of magnitude κ_1 and direction ϕ. We computed zero-crossings and extrema of κ_1 with the data smoothed by Gaussian masks of different variance. These features were then combined to give surface descriptions.

This method worked well on many examples, but exhibited strange behavior in others. The principal cause of the difficulty is that the directions of the two principal curvatures κ_1 and κ_2 can switch when the two have nearly equal values. This happens near a saddle surface, but also due to even small perturbations in a nearly flat surface.

To overcome these problems, our new implementation uses curvature properties along various specific directions (though still computed from the principal curvatures κ_1 and κ_2). The algorithms for tracking across different scales and labelling are also different.

3.2. Description of the new approach

A block diagram of our method is given in figure 1. Our basic approach is to first smooth the image with a number of Gaussian masks of different variance and compute the curvature in four different directions. We then detect extrema and zero-crossings of the resulting curves in each direction and for each smoothed image (i.e. at different scales). After filtering out insignificant features, features from different scales are combined in the step of "space-tracking" to give robust and well-localized features. We then combine adjacent features ("space grouping") followed by merging of results from the four directions. We now have point features of interest, which are linked to form curves of interest. Additional descriptions of these curves finally allow us to assign descriptions in terms of physical properties of the surfaces.

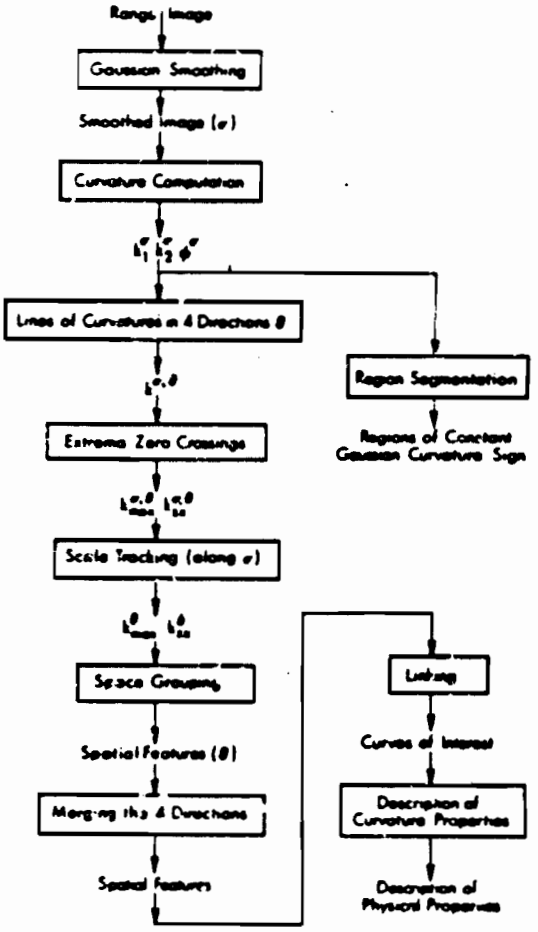


Figure 1: Block diagram of the overall processing

These steps are explained in detail below. To illustrate our steps, we will use the example of a "cup" with an elliptical cross-section shown in figure 2. This data was obtained using an active stereo range finding system at INRIA [23], courtesy of Dr Fabrice Clare. (The elliptical effect was created artificially by scaling the data.) The resolution is 100 x 80 pixels.

3.2.1. Gaussian smoothing

Computation of curvature is likely to be highly noise sensitive. To decrease the effects of noise, we can use a larger support for computing differences, at a possible cost of accuracy of localization. We convolve the image with rotationally symmetric Gaussian masks of different variance, σ. The different sizes of the filter give us curvature at different scales, and are in fact helpful in interpreting the results (as in [24] and [22] for other

applications). The width of the mask is chosen such that the volume under the truncated Gaussian is very close to 1 (6σ is a good approximation). In the current implementation, we use σ=0.5, 1.0, 1.5.

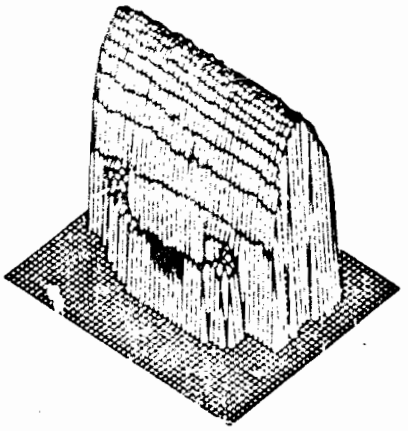


Figure 2: 3-D plot of the "cup" image

3.2.2. Derivatives

Since we work with discrete data, we compute differences rather than derivatives. We compute the first-order differences in the x and y directions, and the second order cross derivative by convolving the images with the masks shown below

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

1	0	-1
0	0	0
-1	0	1

The output of these masks are normalized by the sum of the weights in the mask. Second order difference are obtained by convolving with these masks again.

3.2.3. Computation of Principal Curvatures

Once these derivatives are estimated, the two principal curvatures are computed by solving a second degree equation, as explained in [8]. By convention, κ_1 denotes the larger of the two. The orientation is obtained by solving a first degree differential equation. Since the principal directions are orthogonal, it is sufficient to represent the angle ϕ of κ_1 with the X-axis. We are currently investigating alternative ways to compute the principal curvatures, such as by using the facet model [25]. Figure 3 shows a "needle diagram" computed from the cup data; a line is shown in the direction of the maximal principal curvature, the length of the line is proportional to the magnitude of this curvature.

3.2.4. Gaussian Curvature

Some previous empirical observations suggest that the shape of the regions of constant Gaussian curvature sign roughly reflects the overall shape of an object [26]. We compute these regions from the unsmoothed range image (to preserve accurate localization of boundaries). To get rid of very small regions, we apply a sequence of expand and contract operations. At this stage, we have our first surface description, obtained independently of the subsequent processing applied to the image. The cup figure, however, has no such interesting regions as Gaussian curvature is essentially zero everywhere (another example is shown in the results section later). In figure 4, zones of zero gaussian curvature are shown in grey, positive in black and negative in white.

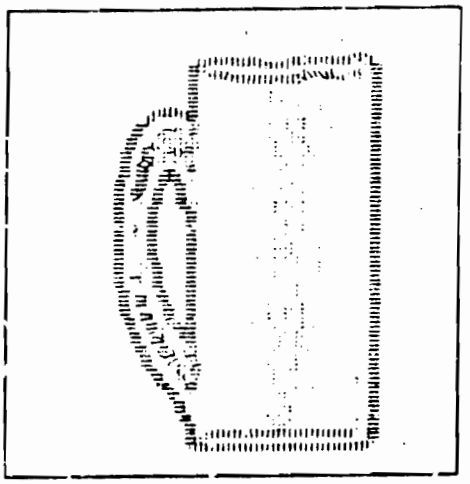


Figure 3: Needle representation of the maximum principal curvature

Figure 4: Regions of constant Gaussian curvature sign

3.2.5. Curvature along a line

Given the principal curvatures κ_1, κ_2 and the angle ϕ, it is possible to compute the curvature in any direction ψ:

$$\kappa_\psi = \kappa_1 \cos^2(\phi-\psi) + \kappa_2 \sin^2(\phi-\psi)$$

Curvature values along a given direction give us a one-dimensional curve. In the current implementation, we use 4 directions ψ = 0°, 45°, 90° and 135°. It may be more accurate to compute these quantities using different masks, or directly from an approximation such as the facet model; we are investigating these issues currently.

3.2.6. Extrema and Zero-crossings

For each of these one dimensional curves, we compute the zero-crossings and the local extrema. A local extremum is defined as a point whose absolute value strictly larger than the absolute value of one of its two neighbors and larger or equal to the other one. Extrema below a small threshold are discarded. A zero-crossing is given by a zero surrounded by non-zero numbers of opposite sign on the two sides or by a sequence of two numbers of opposite sign; in the latter case, its location is marked at the smaller of the two numbers. No thresholding is performed at this stage. Each zero-crossing also has a positive extremum and a negative extremum associated with it on either side.

3.2.7. Filtering

At any given point, only information in one direction is useful, in general. If a jump boundary occurs in the 0° direction, the descriptions in the other directions at the same location should be ignored. A zero-crossing is retained only if one of its associated extrema meets the criterion that, at the extremum, the value of the curvature in directions different from the current direction not exceed the extremum value. Note that it is still possible for a zero-crossing and extrema to be associated with more than one direction after this step.

Figure 5 shows the extrema and zero-crossings detected from the cup data in the X-direction (horizontal). The top row shows the positive extrema, the middle row the zero crossings and the bottom row the negative extrema. The four columns used smoothing masks of variance 0, 0.5, 1.0 and 1.5 from left to right respectively. Figure 6 shows similar results for the vertical direction.

3.2.8. Scale tracking

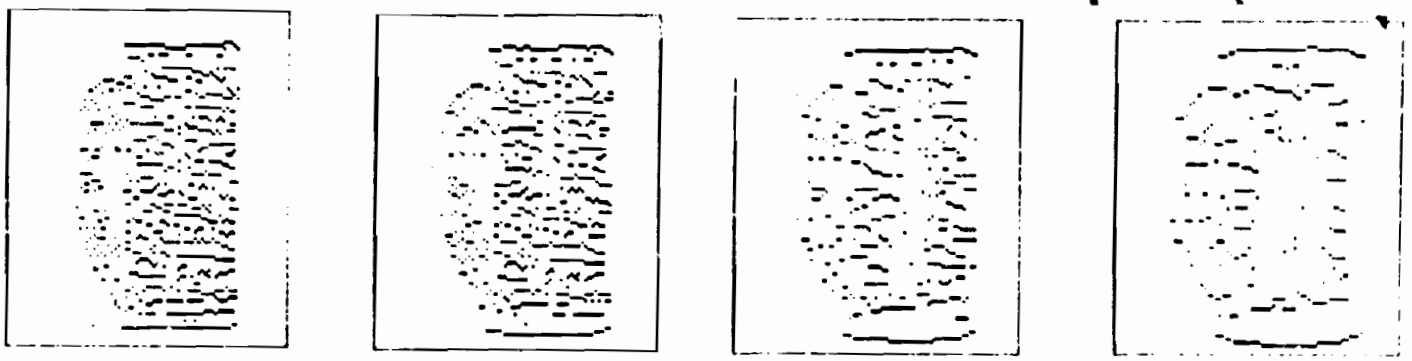
Use of multiple scales allows us to increase our confidence in detecting features without loss of localization. By increasing the scale, we increase the signal-to-noise ratio and therefore the confidence in the feature that we extract, as the cost of accuracy of localization. Here, we detect the features in the image smoothed with the widest filter, and localize them using the smallest filter. Since we are using only a discrete set of filters, we have to solve a correspondance problem between levels. As the scale increases, we know that new features may not appear, but that features from a lower level may merge. Shifts of the features for different scales can also be predicted [7].

In our implementation, we track extrema and zero-crossings independently. The strategy we follow is *coarse-to-fine*: only features present at the coarsest level may be part of the description. (For some applications, it may be advantageous to have a hierarchical description, consisting of features tracked at all levels, then at all levels except the coarsest, and so on.) The search is one-dimensional, and the amount of displacement of a given feature depends on the value of σ (2 pixels at most for a 0.5 variation in σ). The direction of displacement must remain the same for different scales for an extrema, but is allowed to change by one pixel for a zero-crossing (as its localization is ambiguous by one pixel).

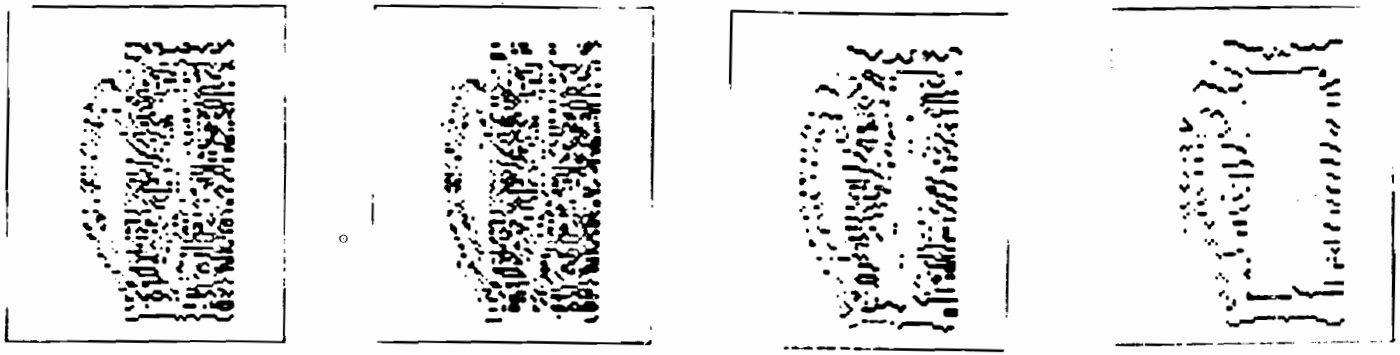
When tracking an extremum, if we come to a "fork", that is a choice between two extrema at the next finer scale, we choose the extremum with the higher curvature value. If we come to a fork for a zero-crossing, we stop the tracking and simply mark the position at the lowest unambiguous level.

At this point we filter out some zero-crossings that have been introduced due to the following artifact. In the cup example, ideally, the curvature is maximum along horizontal curves and zero along the vertical. However, instabilities occur near the edge. Near the edge, the surface is receding rapidly from the viewer and hence the first derivative f' is large (tends to ∞). As curvature is given by $f''/(1+f'^2)^{3/2}$, the curvature tends to zero. Thus, if there is a small 'bump' in the surface near the edge, the curvature in the vertical direction is likely to be higher than in the horizontal direction giving undesirable zero-crossings. This is illustrated in fig. 5(b) which shows many zero-crossings in the Y direction close to the right border of the cup. To handle this, we remove those zero-crossings where the derivative in the orthogonal direction is high (above a threshold).

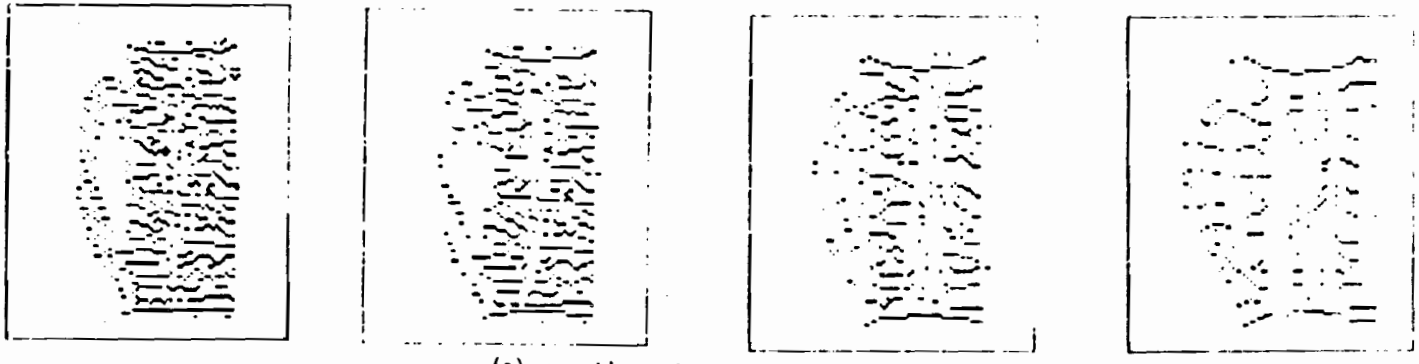
Figure 7 shows the results after scale tracking. The top and bottom rows show the results in the X and Y directions respectively. The first column shows the positive extrema, the second the negative extrema and the last the zero-crossings.



(a) positive extrema

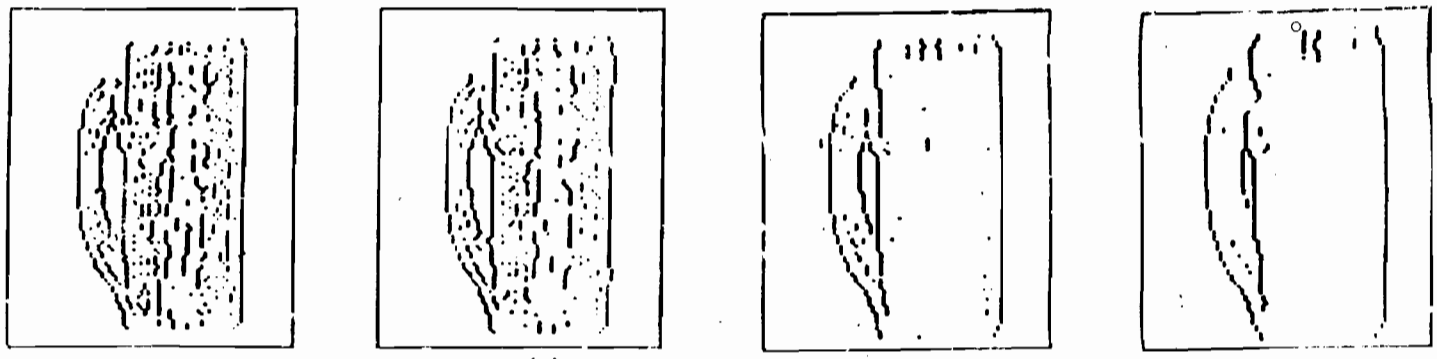


(b) zero crossings

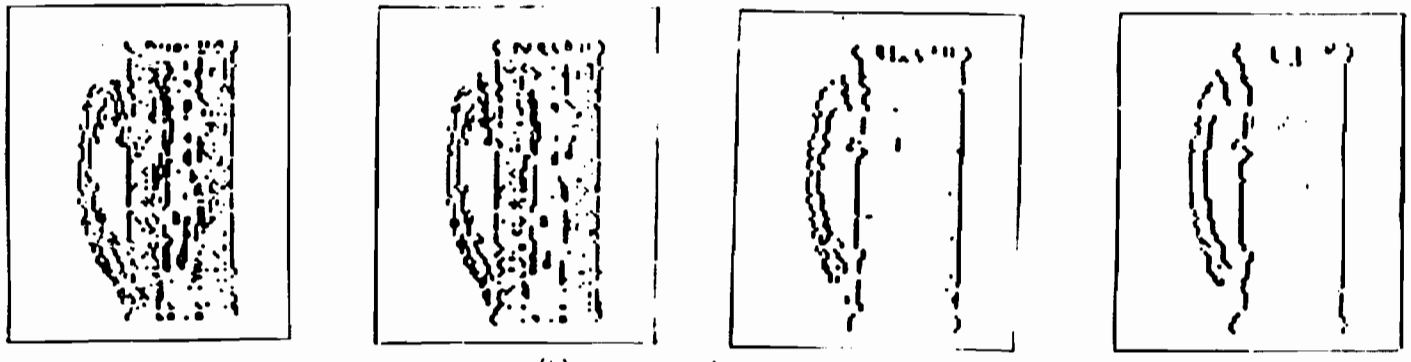


(c) negative extrema

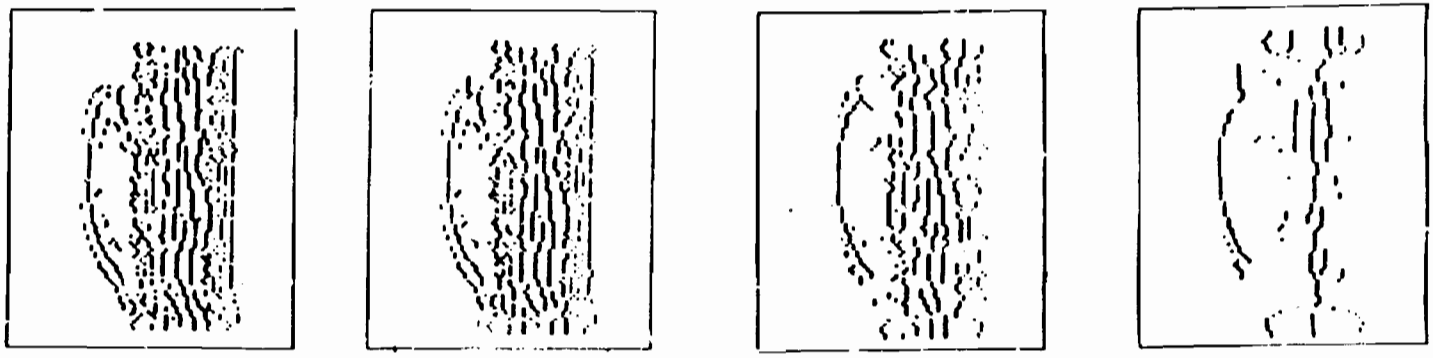
Figure 5: Extrema and zero-crossings in the horizontal direction for σ increasing from 0 to 1.5



(a) positive extrema



(b) zero crossings



(c) negative extrema

Figure 6: Extrema and zero-crossings in the horizontal direction for σ increasing from 0 to 1.5

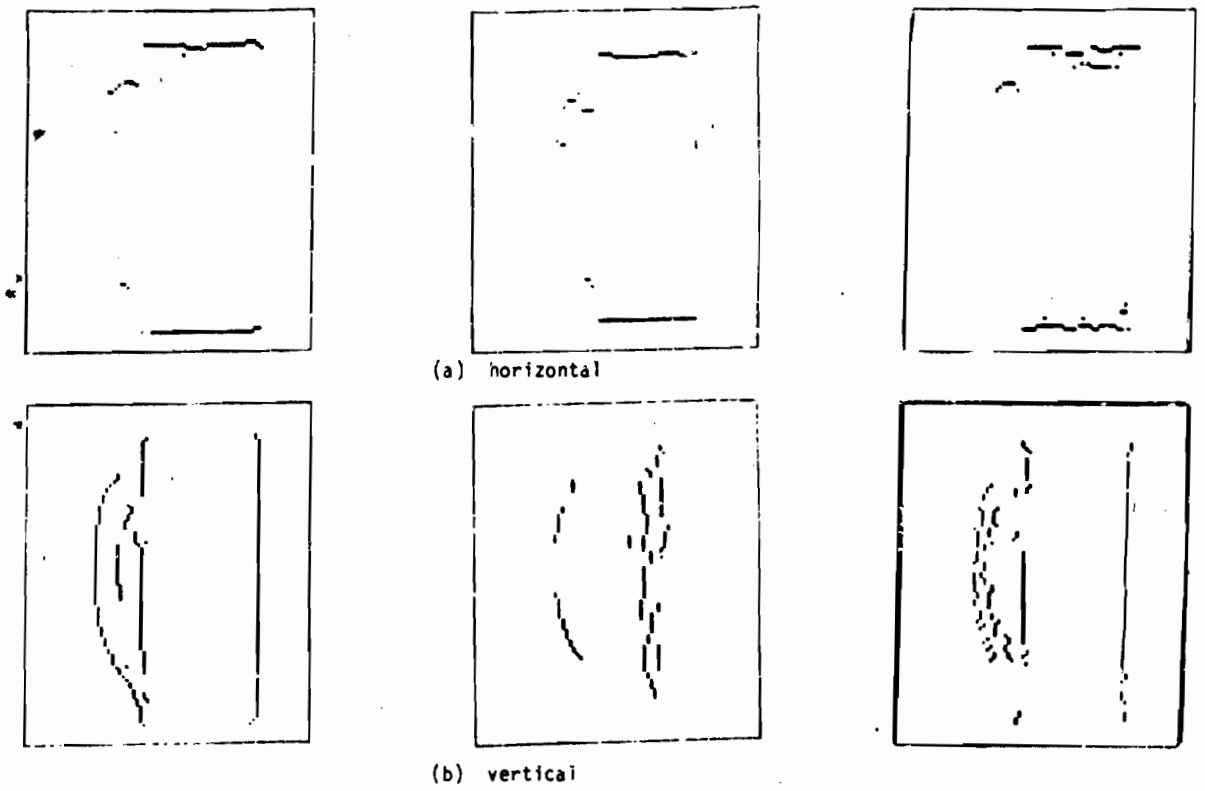


Figure 7: Results of the scale tracking procedure from left to right: positive extrema, negative extrema, zero-crossing

3.2.9. Space grouping

This step is used to find the characteristic groups of features (extrema and zero-crossings) that we call junctions. We define 5 types of junctions

- isolated positive extremum (+)
- isolated negative extremum (-)
- linked positive extremum and zero-crossing (+0)
- linked negative extremum and zero-crossing (-0)
- linked positive extremum and zero-crossing and negative extremum (+0-)

These 5 junctions generate in turn the following dictionary of 8 labels for the individual features of a junction:

label 1: isolated positive extremum

label 2: isolated negative extremum

label 3: zero-crossing linked to a positive extremum only

label 4: zero-crossing linked to a negative extremum only

label 5: zero-crossing linked to a positive and a negative extremum

label 6: isolated zero-crossing (to be discarded)

label 8: positive extremum linked to a zero-crossing (label 3 or 5)

label 9: negative extremum linked to a zero-crossing (label 4 or 5)

The algorithm to compute these junctions is applied for each direction independently (and hence is one-dimensional). We first locate a feature and then search for another feature along the given direction (limited to 3 pixels based on analysis of expected locations). Depending on the type of the adjacent features, labels are assigned to the junction (group of features). (For a zero-crossing, up to three features may need to be labelled simultaneously). The grouping and labelling algorithm is given below:

```
TRACE(p): pixel value of input picture;
JUNCT(p): pixel value of output picture;
  flag(p): flag of point p;
  zero,plus,minus: flags for zero-crossing, positive
    and negative maxima;

  1. For each nonzero point p1 in the input picture do:
  2. Find the first nonzero point p2 in the positive direction
     of searching (+y, +x, +45, +135) and within distance 3:
  3.   case (TRACE(p1),TRACE(p2)):
         (plus,zero):
         (zero,plus):  flag(p1):=flag(p1) OR zero OR plus;
                       flag(p2):=flag(p2) OR zero OR plus;
                       break;
              (minus,zero):
              (zero,minus): flag(p1):=flag(p1)OR zero OR minus;
                            flag(p2):=flag(p2)OR zero OR minus;
                            break;
  4. end 1;
  5. For each nonzero point p in the input picture do:
  6.   Assign JUNCT(p) according to following table:
```

TRACE	flag	JUNCT
plus	0	1
minus	0	2
zero	zero+plus	3
zero	zero+minus	4
zero	zero+plus+minus	5
zero	0	6
plus	zero+plus	8
minus	zero+minus	9

3.2.10. Merging

The four junction images are merged into a single image. Each pixel is again labelled according to the dictionary given above. It is possible for a pixel to have been assigned a label in more than one direction. If so, we choose the label with the highest priority as defined by its position in the following list:

5	zero-crossing with + a
3,4	zero-crossing with + o
8,9	+ or - with zero-cross
1,2	single + or -
6	single zero

If a pixel is marked with labels having the same priority level, we keep the one having the largest magnitude. Note that due to digital computations, the same feature may also appear at adjacent pixels in different directions. In this case, we used a simple "thinning" process, as described in [27].

Figure 8 shows the results of space grouping on the cup data. Fig. 8(a) shows the zero-crossings (labels 3, 4 and 5), fig. 8(b) shows the isolated negative extrema (label 2) and fig. 8(c) the isolated positive extrema (label 1). Figure 8(d) shows all of these together. In particular, note that the edges in the middle of fig. 8(b) correspond to the distinguished points on the elliptical cross-section of the cup.

3.2.11. Spatial linking

The objective of this step is to connect point features to form curves. First, we localize each junction at the position of the zero-crossing if the junction has one (+0,-0 or +0-), at the extremum location if it is isolated (+ or -). Junctions with the same label are linked if their orientation is compatible (45° or less apart), and one-pixel gaps are filled. If we come to a fork, we choose the longest branch (found by look-ahead search), and generate separate lines for the smaller branches. At the end of this step, we have a non-iconic description of the surface.

3.2.12. Curvature descriptors

We can now associate additional descriptions with the detected curves as follows:

- isolated extrema:

  the value of the maximum principal curvature and its orientation are sufficient .

- extrema - zero-crossing:

  locally, this is represented by the value of the extremum of curvature κ_1 and two angles α and β as shown in figure 9.

- extremum zero-crossing extremum:

  This represented by the values of the two opposite sign extrema and the four angles α_1, α_2, β_1, β_2 as shown in figure 10.

An example of the resulting description is shown below for the cup image:

```
Line #1:
Starting point (21,63);Ending point (84,63)
  Line length = 64;  Type = 3
  Strength = 7.634127; Alpha 1 = 11.222473;
    Alpha2 = -3.654785;
  Beta1 = 42.866863;  Beta2 = 42.866863

Line #5:
Starting point (6,36); Ending point (6,61)
 Line length = 26;  Type = 5
 Strength = 20.041059; Alpha1 = 6.844843;
   Alpha2 = -1.430612;
 Beta1 = 12.753203;  Beta2 = =1.742495
```

3.2.13. Surface descriptors

We can now translate the curvature descriptions to descriptions of significant surface changes.

- jump boundaries:

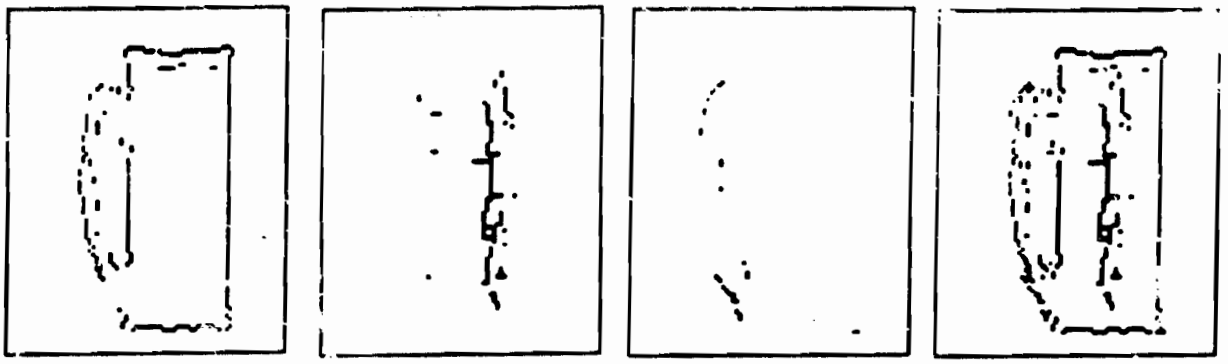
  they correspond to the plus-zero-minus above. The same type of descriptions can be derived (height of the step and orientation of the two surfaces).

- folds:

  they correspond to the extremum-zero above. They need to be further classified to describe the 2 adjoining surfaces, and the orientation of the fold with respect to the viewer.

- curvature extrema:

they are the single extrema. We need to give the orientation with respect to the viewer.



(a) zero-crossings (b) isolated negative maxima (c) isolated positive maxima (d) all features

Figure 8: Results of the space grouping procedure

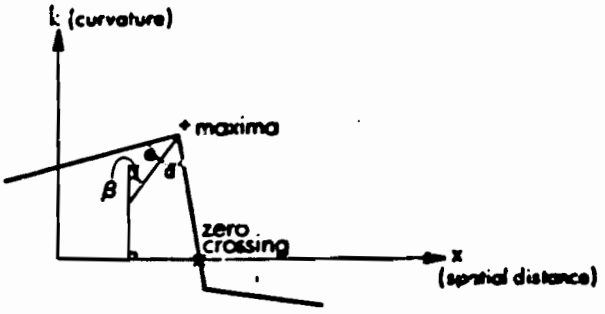


Figure 9: Curvature descriptors for +0 junction

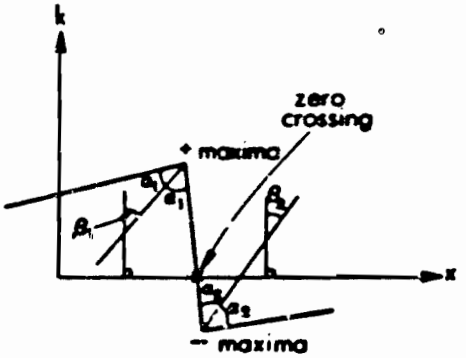


Figure 10: Curvature descriptors for a +0- junction

4. RESULTS AND CONCLUSIONS

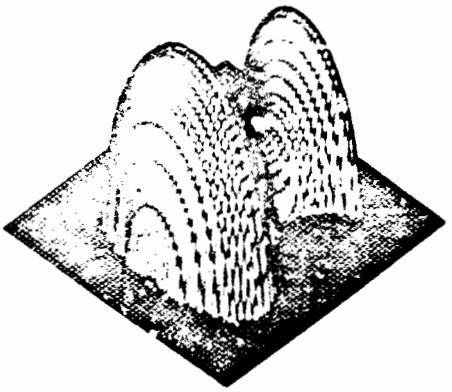
We show results on one more example here (the processing steps are the same as in the previous section). Fig. 11(a) shows the object (it is a synthetic range image of a half-bottle). Figure 11(b) shows the needle diagram and fig. 11(c) shows regions of positive and negative curvature (lighter region shows negative Gaussian curvature regions, darker regions show positive Gaussian curvature regions). For this example, the Gaussian curvatures gives a useful crude segmentation.

Figure 12 shows the main results of our processing. Figure 12(a) shows the zero-crossings, fig. 12(b) the isolated positive extrema, fig. 12(c) the isolated negative extrema and fig. 12(d) shows them all together.

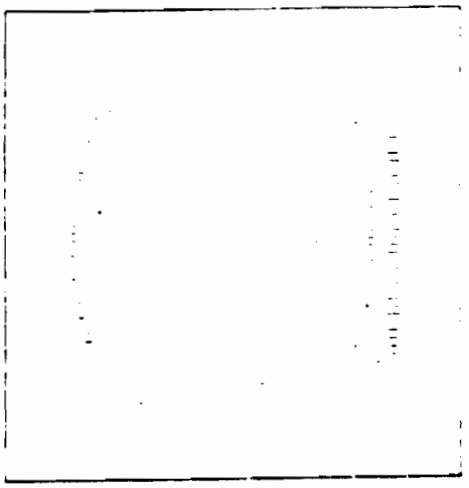
We can draw some conclusions from these two examples. First, significant occluding boundaries and fold boundaries are detected well (though jump boundaries would be detected as well or better by any normal edge detector also). In addition, other significant curves are also detected on the cup, the line in the middle corresponds to the apex of the elliptical cross-sections. In the bottle, we detect the curvature extrema where the bottle cross-section is the largest and also where it is the smallest.

We have used these descriptions for the bottle to reconstruct the original surface using an implementation by Cochran [28] at USC that follows Terzopoulous' scheme [26, 29] as shown in figure 13.

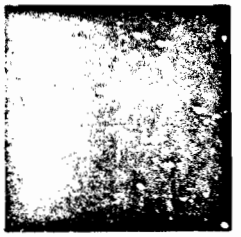
We believe that these results show the essential utility and feasibility of the proposed representation scheme, though many of the details can be improved. We are in the process of testing our methods on a larger set of images and using the descriptions for higher levels of processing.



(a) 3-D plot of the "bottle" image

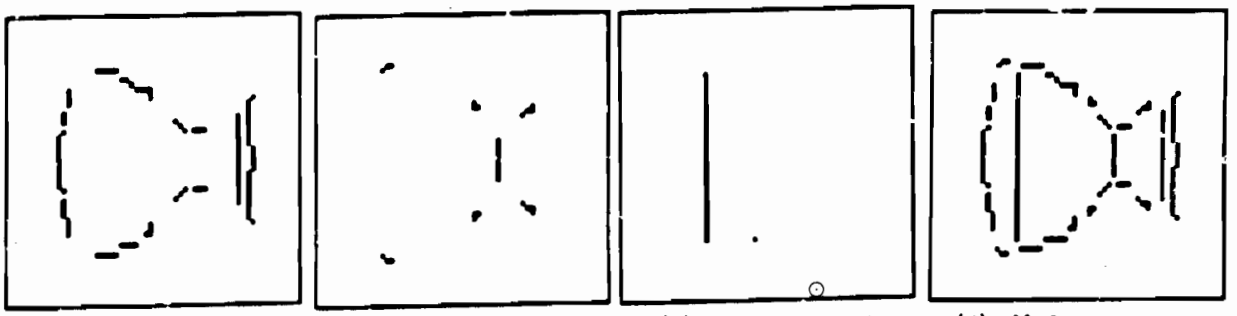


(b) Needle representation of the maximum principal curvature



(c) Regions of constant Gaussian curvature sign

Figure 11:



(a) zero-crossings (b) isolated positive maxima (c) isolated negative maxima (d) all features

Figure 12: Results of the space grouping procedure

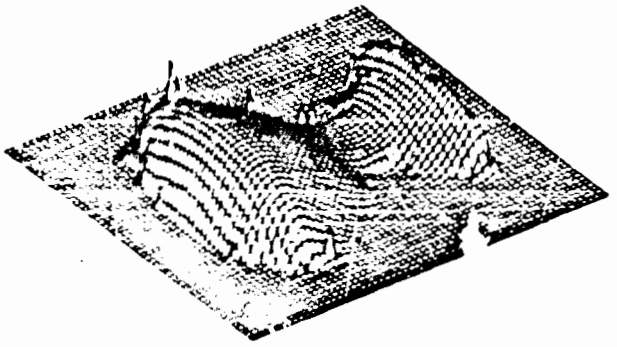


Figure 13: Reconstruction of the "bottle" image from descriptors generated by the program

References

1. Binford, T.O., "Visual Perception by Computer," *IEEE Conference on Systems and Controls*, December 1971.

2. Nevatia, R. and Binford, T.O., "Description and Recognition of Complex-Curved Objects," *Artificial Intelligence*, Vol. 8, 1977, pp. 77-98.

3. Marimont, David H., "A Representation for Image Curves," *AAAI-84*, 1984, pp. 237-242.

4. Brady, M., Ponce, J., Yuille, A. and Asada, H., "Describing Surfaces," *Proceedings of the 2nd International Symposium on Robotics Research*, H. Hanafusa and H. Inoue, eds., Massachusetts Institute Technology Press, Cambridge Mass., 1985.

5. Besl, P.J. and Jain, R.C., "Intrinsic and Extrinsic Surface Characteristics," *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, San Francisco, Calif., June 9-13 1985, pp. 226-233.

6. Sethi, I.K. and Jayaramamurthy, S.N., "Surface Classification using Charac[illegible] Contours," *Proceedings of Inter[illegible] Conference on Patterr[illegible]* August 1984, pp. 438-440.

7. Ponce, J., and Brady, M., "[illegible] Primal Sketch," *Proceedings [illegible] International Conferenc[illegible] Robotics and Automation*, St Louis, [illegible]arch 25-28 1985, pp. 420-425.

8. Medioni, G. and Nevatia R., "Matching Images Using Linear Features," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 6, No. 6, November 1984, pp. 675-685.

9. Besl, P.J. and Jain, R.C., "Three-Dimensional Object Recognition," *ACM Computing Surveys*, Vol. 17, No. 1, March 1985, pp. 75-145.

10. Milgram, D.L. and Bjorklund, C.M., "Range Image Processing: Planar Surface Extraction," *International Joint Conference on Pattern Recognition-5*, 1980, pp. 912-919.

11. Henderson, T.C., "Efficient 3-D Object Representations for Industrial Vision Systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 5, No. 6, November 1983, pp. 609-617.

12. Bhanu, B., "Surface Representation and Shape Matching of 3-D Objects," *Proceedings of the IEEE Pattern Recognition and Image Processing Conference*, Las Vegas, Nev., June 14-17 1982, pp. 349-354.

13. Hebert, M. and Ponce J., "A New Method for Segmenting 3-D Scenes into Primitives," *International Joint Conference on Pattern Recognition*, 1982, pp. 836-838.

14. Oshima, M. and Shirai, Y., "A Scene Description Method Using Three-Dimensional Information," *Pattern Recognition*, 1979, pp. 9-17.

15. Inokuchi, Seiji, Nita, Takashi, Matsudew, Fumio, Sakurai, Yoshifumi, "A Three-Dimensional Edge-Region Operator for Range Pictures," *Proceedings of International Joint Conference on Pattern Recognition-6*, October 1982, pp. 918-920.

16. Mitiche, A. and Aggarwal, J.K., "Detection of Edges Using Range Information," *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 5, No. 2, March 1983, pp. 174-178.

17. Bolles, R.C., Horaud, P. and Hannah, M.J., "A Three-Dimensional Part Orientation System," *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, August 8-12 1983, pp. 1116-1120.

18. Lin, C. and Perry, M.J., "Shape Description using Surface Triangularization," *IEEE Proceedings Workshop on Computer Vision: Representation and Control*, August 1982, pp. 38-43.

19. Laffey, Thomas J., Haralick, Robert M., Watson, Layne T., "Topographic Classification of Digital Image Intensity Surfaces," *IEEE Proceedings Workshop on Computer Vision: Representation and Control*, August 1982, pp. 171-177.

20. Langridge, D.J., "Detection of Discontinuities in the First Derivatives of Surfaces," *Computer Vision, Graphics, and Image Processing*, Vol. 27, September 1984, pp. 291-308.

21. Nackman, L.R., "Two-Dimensional Critical Point Configuration Graphs," *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 6, No. 4, July 1984, pp. 442-449.

22. Witkin, A.P., "Scale-Space Filtering," *Proceedings of Seventh IJCAI*, Karlsruhe, West Germany, August 1983, pp. 1019-1022.

23. Boissonnat, J.D. and Faugeras, O.D., "Triangulation of 3-D Objects," *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver, B.C., Canada, August 24-28 1981, pp. 658-660.

24. Asada, H. and Brady, M., "The Curvature Primal Sketch," *Proceedings of the 2nd IEEE Workshop on Computer Vison: Representation and Control*, Annapolis, MD, May 1984, pp. 8-17.

25. Haralick, R.M., "Digital Step Edge from Zero-Crossings of Second Directional Derivatives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 1, January 1984, pp. 58–68.

26. Terzopoulos, D., *Multiresolution Computation of Visible-Surface Representations*, PhD dissertation, Massachusetts Institute of Technology, Departments of Computer Science and Electrical Engineering, January 1984.

27. Rosenfeld, A. and Kak, A., *Digital Picture Processing*, Academic Press, New York, 1977.

28. Cochran, S. and Medioni, M., "Implementation of a Multiresolution Surface Reconstruction Algorithm," to appear in a USC internal technical report

29. Terzopoulos, D., "Computing Visible Surface Representations," *Massachusetts Institute Technology AI Lab*, No. AI Memo 800, 1985..

GC970145-30

DISPARITY FUNCTIONALS AND STEREO VISION

Roger D. Eastman and Allen M. Waxman*

Center for Automation Research
University of Maryland
College Park, Maryland 20742

ABSTRACT

This paper investigates stereo matching constraints that derive from an analytic model of surface depth. Computational stereo is formulated as a single stage process in which potential feature point or contour matches interact to provide support for local estimates of a polynomial model of disparity (the *disparity functional*), not just estimates of disparity at isolated points. An algorithm is presented that integrates the disparity functional with multiresolution matching of zero-crossings to derive depth to surface patches. The analyticity of the disparity field is thereby exploited early in the matching process, and yields surface reconstruction as a direct byproduct of correspondence.

1. INTRODUCTION

Recent computational approaches to the problem of stereo correspondence have emphasized the use of geometric matching constraints derived from models of camera optics, relative camera geometry, scene depth and photometry. Smoothness is an important property of depth that can be translated into a matching constraint. Since depth usually varies smoothly across surfaces, the disparity given by correct matches should also vary smoothly except at surface boundaries (Marr and Poggio [8]).

In this paper, we investigate matching constraints that derive from an analytic model of surface depth combined with a model of parallel stereo cameras. Analyticity mathematically formulates smoothness by modeling object surfaces, and therefore the disparity field, as piecewise analytic functions of visual direction. Our model of *analytic coherence* mathematically formulates the principle of *coherence* stated by Prazdny [12], and can describe transparent as well as opaque surfaces. In using this property, we follow the work in stereo of Koenderink and van Doorn [6] and the work in motion of Waxman and Ullman [19], Waxman [17], Waxman and Wohn [20] and Wohn [21]. Waxman and Wohn [20] developed the *Velocity Functional Method* for recovering the deformation parameters of image flow fields, and we propose here the *Disparity Functional Method* for the disparity field. The *disparity functional* is a polynomial model of the disparity field in a neighborhood. We show in this paper that the locally linear disparity functional is a useful representation of the disparity field for performing both correspondence and surface reconstruction.

We formulate stereo as a single stage process in which potential feature point and contour matches interact to provide local support for estimates of the disparity functional coefficients (and therefore local surface structure), not just estimates of disparity at isolated points. This extends the notion of local support defined by Marr and Poggio [8]. Those matches that do not participate in good estimates would fail to win local support and be eliminated, while the computed coefficients give the local variation of depth; no extensive surface reconstruction is needed. This formulation rests on the principle that locally consistent image deformations are strong evidence for a locally smooth surface. The approach relates to the *Binocular Raw Primal Sketch* principle of Mayhew and Frisby [9], which proposes that the construction of extended image primitives (i.e., contours) should occur simultaneously with the construction of disparity field primitives (i.e., surface patches.) We propose an algorithm that computes the local disparity functional from contour matches. As with feature point matches, those contours that participate in good matches can be preserved, those that do not can be discarded.

We present in this paper two algorithms for recovering the locally linear disparity functional. One of the algorithms integrates the linear disparity functional into the Marr-Poggio-Grimson matching algorithm (Grimson [4]) as a measure of local support; the other uses the linear functional as a measure of contour correspondence. In Section 2, we analyze relationships between models of viewpoint geometry, models of depth and models of the disparity field. In Section 3, we use these relationships to derive the principles of the disparity functional method and we present preliminary implementations of the method.

*Present address: Thinking Machines Corporation, 245 First Street, Cambridge, MA 02142 .

2. THE DISPARITY FIELD

In applying the smoothness property to transparent surfaces, Prazdny [12] stated the following principle of coherence: "A discontinuous disparity field may be a superposition of several interlaced continuous disparity fields, each corresponding to a piecewise smooth surface." We wish to state a strong mathematical formulation of coherence, that depth (and therefore disparity) can be modeled as *overlapping analytic regions* (essentially piecewise C^1) with singularities at opaque occluding boundaries or sharp changes in orientation; a region need not determine actual scene depth for all visual directions that it subtends, but each visual direction is associated with only one region and therefore one depth. This principle of *analytic coherence* follows from the principle of *flow analyticity* used by Waxman and Wohn [20], and establishes a model for the disparity field.

The piecewise analyticity of disparity allows us to examine its structure in a small neighborhood of an image point by Taylor series expansion, and to relate the terms of the Taylor series to relative image deformation and surface structure. By fitting a polynomial model to disparity in a neighborhood, we can recover the Taylor series coefficients and thereby both image deformation and surface depth. It is this polynomial model that we define as the *local disparity functional*. For the sake of simplicity, we have restricted the geometric analysis in this paper to the case of stereo cameras with parallel optic axes and image planes. This makes the relationships between the disparity functional and surface structure very straightforward, and also simplifies the search process since disparity becomes a scalar field. This simplification avoids the need to recover generally unknown vergence and gaze angles of the two cameras (the *interpretation* problem.) In the complex case, disparity is a vector field and the terms of the disparity functional compound surface structure with the vergence and gaze. The analysis we present here follows from the analyses of stereo by Koenderink and van Doorn [6], Mayhew and Longuet-Higgins [10] and those of motion by Longuet-Higgins and Prazdny [7], Waxman and Ullman [19] and Waxman and Wohn [20]; see also Grimson [4].

The disparity functional models the local structure of the disparity field. The disparity field of a natural scene will have a global structure that reflects regions where the underlying surface is analytic, intervening occluding boundaries and structural edges, and in general the topology of the visible scene. This must be modeled by a segmentation of the disparity field into regions where the coefficients of the disparity functional vary smoothly, a process discussed in the context of motion in Waxman [17] and Waxman and Wohn [20], and in the context of combined motion and stereo analysis in Waxman and Duncan [18]. Transparent surfaces complicate this model, since a single neighborhood can contain multiple interlaced disparity fields. Eastman and Waxman [3] discusses the use of multiple disparity functionals in a neighborhood to model this case.

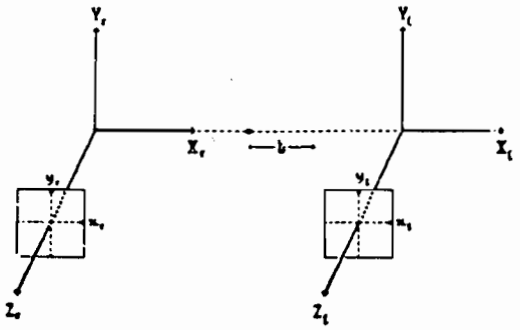


Fig. 1 - Stereo Camera Model with Coordinate Systems

Our objective in the rest of this section is to define the disparity functional for a small image neighborhood, and to relate the terms of the functional to local image deformations and underlying surface structure. We adopt the notation of upper case for variables in world coordinates and lower case for image coordinates. We assume a camera model consisting of two pin-hole cameras with parallel optic axes and coincident image planes. The focal point of the left camera is located at the origin of a right handed coordinate system (X_l, Y_l, Z_l), while the right camera defines a similarly oriented coordinate system (X_r, Y_r, Z_r) with its origin at $(-B, 0, 0)$ in the left camera system; the systems are illustrated in Figure 1. This gives a stereo baseline of B and the systems are related by the equation $(X_r, Y_r, Z_r) = (X_l + B, Y_l, Z_l)$. The positive Z-axis for each system is directed along the line of sight. The image coordinate systems (x_l, y_l) and (x_r, y_r) are normal to their respective Z-axes with their origins at $Z = 1$, so under perspective projection the images are reinverted and scaled to a focal length of unity. A single point in the world, (X_l, Y_l, Z_l), projects into the left and right images, respectively, at

$$(x_l, y_l) = \left(\frac{X_l}{Z_l}, \frac{Y_l}{Z_l}\right) \quad , \quad (x_r, y_r) = \left(\frac{X_l - B}{Z_l}, \frac{Y}{Z_l}\right) \qquad (1a,b)$$

This yields equations for horizontal and vertical disparity (in the left image coordinates) of

$$\delta_x(x_l, y_l) = x_r - x_l = \frac{B}{Z_l(x_l, y_l)} \quad , \quad \delta_y(x_l, y_l) = y_r - y_l = 0 \qquad (2a,b)$$

At this point, the right coordinate systems are superfluous since we can write everything in terms of the left systems (or cyclopean systems); we intend to do this, and therefore drop the subscripts r and l. Also, further references to disparity concern only $\delta = \delta_x$.

We now combine the camera model with an analytic model of depth. Consider a small neighborhood in the left image centered at the origin $(x, y) = (0,0)$. We assume that depth $Z(X, Y)$ is analytic and single valued along the line of sight in this neighborhood, so we can expand it in a Taylor series about $(X, Y) = (0,0)$:

$$Z = Z_0 + \left.\frac{\partial Z}{\partial X}\right|_{0,0} X + \left.\frac{\partial Z}{\partial Y}\right|_{0,0} Y + \frac{1}{2}\left.\frac{\partial^2 Z}{\partial X^2}\right|_{0,0} X^2$$

$$+ \frac{1}{2}\left.\frac{\partial^2 Z}{\partial Y^2}\right|_{0,0} Y^2 + \left.\frac{\partial^2 Z}{\partial X \partial Y}\right|_{0,0} XY + \cdots \qquad (3)$$

To second order, this defines a quadric surface patch with two slopes and three curvatures. To abreviate the equations, we will rename the partials as follows: $P = \left.\frac{\partial Z}{\partial X}\right|_{0,0}$, $Q = \left.\frac{\partial Z}{\partial Y}\right|_{0,0}$, $C_{XX} = \left.\frac{\partial^2 Z}{\partial X^2}\right|_{0,0}$, $C_{YY} = \left.\frac{\partial^2 Z}{\partial Y^2}\right|_{0,0}$ and $C_{XY} = \left.\frac{\partial^2 Z}{\partial X \partial Y}\right|_{0,0}$.

Since disparity is directly proportional to the reciprocal of depth $z = 1/Z$ (and depth is positive definite), the local analyticity of depth implies the analyticity of disparity. However, we need to convert (3) to refer to reciprocal depth as a function of image position (x, y):

$$\frac{1}{Z} = \frac{1}{Z_0}(1 - Px - Qy)$$

$$- (\frac{1}{2}C_{XX}x^2 + \frac{1}{2}C_{YY}y^2 + C_{XY}xy) + O(x^3). \qquad (4)$$

Equation (4) is the desired relationship between reciprocal depth and image postion; its derivation follows from Waxman and Ullman [19].

We can now substitute (4) into the equation for disparity (2a) to get a functional approximation for horizontal disparity.

$$\delta(x, y) = \frac{B}{Z_0}(1 - Px - Qy)$$

$$- B(\frac{1}{2}C_{XX}x^2 + \frac{1}{2}C_{YY}y^2 + C_{XY}xy) \qquad (5)$$

For image neighborhoods corresponding to curved surface patches, this second order approximation is *locally valid* with an accuracy depending on the curvatures of the patch. For the simple parallel camera model, the rotational terms of vergence, gaze and cyclotorsion do not appear in the disparity equations. As a result, for neighborhoods corresponding to planar surface patches all terms beyond first order vanish and a linear functional approximation is *globally valid*. In this case, horizontal disparity can be written as

$$\delta(x_l, y_l) = \frac{B}{Z_0}(1 - Px - Qy). \qquad (6)$$

If the surface is planar, then *reciprocal depth* and *disparity* are linear functions of image coordinates. This relation was used by Castan and Shen [2], who observed that Z (depth in the world) is then a hyperbolic function of image coordinates; it also follows from the planar model used by Mayhew and Longuet-Higgins [10].

We call this second order polynomial approximation to disparity the *disparity functional* and define its terms as follows:

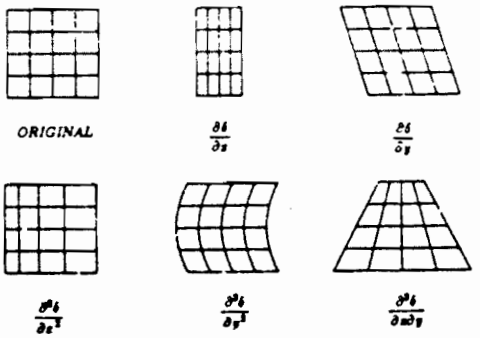


Fig. 2 - Deformations of the Horizontal Disparity Field Under the Parallel Camera Model

$$\delta(x, y) = a + bx + cy + dx^2 + ey^2 + fxy. \qquad (7)$$

These terms relate directly to the local Taylor series of disparity (8), as well as the Taylor series of depth. These relations are given in (9) below.

$$\delta(x, y) = \delta_0(0,0) + \left.\frac{\partial \delta}{\partial x}\right|_{0,0} x + \left.\frac{\partial \delta}{\partial y}\right|_{0,0} y + \frac{1}{2}\left.\frac{\partial^2 \delta}{\partial x^2}\right|_{0,0} x^2$$

$$+ \frac{1}{2}\left.\frac{\partial^2 \delta}{\partial y^2}\right|_{0,0} y^2 + \left.\frac{\partial^2 \delta}{\partial x \partial y}\right|_{0,0} xy + \cdots \qquad (8)$$

$$a = \delta_0(0,0) = \frac{B}{Z_0}, \quad b = \left.\frac{\partial \delta}{\partial x}\right|_{0,0} = -\frac{B}{Z_0}P \qquad (9a,b)$$

$$c = \left.\frac{\partial \delta}{\partial y}\right|_{0,0} = -\frac{B}{Z_0}Q, \quad d = \frac{1}{2}\left.\frac{\partial^2 \delta}{\partial x^2}\right|_{0,0} = -\frac{1}{2}BC_{XX} \quad (9c,d)$$

$$e = \frac{1}{2}\left.\frac{\partial^2 \delta}{\partial y^2}\right|_{0,0} = -\frac{1}{2}BC_{YY}, \quad f = \left.\frac{\partial^2 \delta}{\partial x \partial y}\right|_{0,0} = -BC_{XY} \quad (9e,f)$$

We can restrict acceptable values for the disparity functional coefficients by using a disparity gradient limit. If we favor lower values of $\sqrt{b^2 + c^2}$ (e.g., by thresholding this magnitude at 1), then we are implicitly favoring surface reconstructions closer to the fronto-parallel plane. This is a relative and isotropic disparity gradient constraint. (Koenderink and van Doorn [12], Arnold and Binford [1].)

Following the work of Koenderink and van Doorn [6] and Waxman and Ullman [19], but particularly the presentation in Wohn [21,23] for image flow deformations, we can interpret the terms of the disparity functional as a local transformation between the left and right images. Essentially, we have chosen the set $\{ 1, x, y, x^2, y^2, xy \}$ as a (non-orthogonal) basis for the transformation. This basis is illustrated in Figure 2; other bases are described in [21,23]. In modeling the more physiologically plausible camera geometry of cyclotorsion, Koenderink and van Doorn [6] used an alternative basis. In this case, δ_y is non-zero, and they defined the transformation in the

cyclopean coordinate system located midway between the eyes. Their basis characterized the local transformation to first order as a translation, rotation and deformation (compression and stretch along orthogonal axes). They showed that the deformation component is proportional to the gradient of reciprocal depth (i.e., a simple function of slant and tilt) and is invariant to changes in fixation point.

3. THE DISPARITY FUNCTIONAL METHOD

Most theories of stereopsis divide the necessary computations into two distinct stages: establishing feature correspondence and reconstructing surface structure. This division has important implications for matching constraints based on models of scene depth. In order to apply these constraints, some sort of surface reconstruction must be performed during matching to evaluate how closely the proposed matches fit the surface model. The extent of this "correspondence reconstruction" varies from algorithm to algorithm, but it rarely goes beyond the heuristic selection of an optimal "needle" or "wireframe" structure for raw disparity values at isolated feature points (the raw 2.5-D sketch of Marr), the full reconstruction phase must then interpolate smooth surfaces over the wireframe (the full 2.5-D sketch of Marr, cf. Grimson [4] and Terzopoulos [15]). As an alternative, we present efforts towards a theory of stereo which integrates the two stages of correspondence and reconstruction. The alternative is a ***single stage computation*** which selects those matches and disparity interpretations which directly, not indirectly through heuristics, give the best locally smooth surface reconstruction.

Image feature points that arise from texture markings sample the analytic structure of the underlying surface. Given a set of potential feature point matches in a neighborhood, we can estimate the coefficients of the local disparity functional by least squares. If the set of potential matches is correct, the local disparity functional should be a good fit. If the set contains a significant number of incorrect matches, the fit should be poor. The least squares residual thus serves as a measure of local support.

If a good fit cannot be found in a neighborhood, this is evidence that the local surface may be rough at that scale, there are occluding edges in the neighborhood, or there are multiple transparent surfaces. We list three techniques for dealing with these problems. The first is to fit the disparity functional at multiple scales. The functional serves to approximate the disparity field in a neighborhood; this approximation may be valid at one scale but not at another. An example would be a field of grass, which is smooth at a large scale but rough at a fine scale (the stereo pair in Figure 5 exhibits this property). This shows that image deformations take place at various scales and may need to be similarly recovered. The second is to fit the disparity functional to multiple overlapping neighborhoods, and use a modified split-merge approach to locating disparity discontinuities and improving the fit. The modification comes from using overlapping neighborhoods, so that the surfaces fit to adjacent neighborhoods can be compared in the region of overlap (cp. ***overlap compatibility*** in Waxman and Wohn [20], Waxman and Duncan [18]). The third is to fit multiple disparity functionals to a single neighborhood. As Prazdny [12] noted, transparent (or, more properly, intermittent) surfaces can result in interlaced disparity fields; an example would be a chain-link fence a short distance in front of a brick wall. Disentangling the surfaces requires identifying the two planar disparity functionals that fit subsets of the feature points.

Using the disparity functional residual as a measure of local support rests on the principle that locally consistent image deformations are strong evidence for a locally smooth surface. Other stereo matching algorithms that have emphasized neighborhood deformations include those of Koenderink and van Doorn [6], Prazdny [12], Pollard, Mayhew and Frisby [11] and Quam [13]. The algorithm of Koenderink and van Doorn [6] extracts the deformation component of the disparity field by a comparison of local texture measures in the two images; this suggests an attractive though unproven approach to computing the local disparity functional without establishing feature point correspondences, like the ***diffrequency stereo*** of Tyler and Sutton [16] or the approach to motion of Kanatani [5]. The algorithms of Prazdny [12] and Pollard, Mayhew and Frisby [11] select, by a pairwise voting scheme between feature point matches, the disparity at a point that that minimizes the first order deformation in the surrounding neighborhood; this measure of local support avoids explicitly calculating the local deformation. The algorithm of Quam [13] uses a model of depth for ***warping***, or deforming, a neighborhood for intensity correlation; however, the model is expressed in world, not image coordinates.

We have implemented two algorithms which demonstrate the use of the linear disparity functional for correspondence and reconstruction. The first algorithm uses contours as the matching primitives. A single match of two contours, at least one non-linear, provides enough structure to compute a disparity functional. We select those contour matches that yield low residual functionals and group them to expand the scope of the approximation. This is similar to the measures of contour similarity devised by Suh and Kang [14], and to the algorithm discussed in Yuille and Poggio [24]. The second algorithm uses individual zero-crossings as the primitives. The left image is divided into neighborhoods, and each neighborhood undergoes a search for the disparity functional that best fits a subset of the local zero crossing matches. This algorithm is a variation on the multiresolution zero-crossing matching algorithms of Marr-Poggio-Grimson (Grimson [4]); the major difference is its use of the disparity functional to compute both the local structure of depth and a measure of local support. The algorithm is also similar to the multiresolution correlation algorithm of Quam [13].

Both algorithms use a least squares procedure to fit either a first order or second order polynomial $\delta = p(x, y)$ to a set of n points $\{(x_i, y_i, \delta_i)\}$, where (x_i, y_i) are in left image coordinates and $\delta_i = x_r - x_l$. The average residual error is calculated as

$$E = \frac{1}{n} \sum_{i=1}^{n} | \delta_i - p(x_i, y_i) | \qquad (14)$$

3.1 Contour-based algorithm

Contours are defined as weakly monotonically decreasing eight-connected chains of edge points. This means that extended image contours are broken at vertical minima and maxima, and each contour intersects a horizontal scan line only once. After this processing, each image is represented by sets of contours which are in turn represented as sequences of points. For each contour in the left image, the matching algorithm attempts to find a contour in the right image that best satisfies a measure of similarity based on the linear disparity functional.

STEP 1 Contours as defined above, with a minimum length of 8 pixels, are extracted from the left and right images. Horizontal runs of edge points are approximated by the pixel closest to their midpoint.

STEP 2 If a pair of contours share at least 5 rows, and disparity is positive along the contours, it is marked as a possible match.

STEP 3 For each possible match of these contours, a linear disparity functional is fit by least squares and the residual computed.

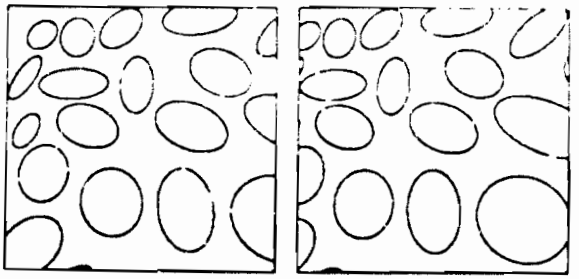
STEP 4 If the linear coefficients yield a disparity gradient above 1, the match is rejected.

STEP 5 For each contour in the left image, the possible matches are ranked by the magnitude of the residual. Matches with a residual more than twice the minimum are rejected. If all matches but the minimum have been eliminated, it is accepted.

STEP 6 If a right contour now participates in an accepted match, the other ambiguous matches it participates in are rejected. This may leave a left contour with a unique match, which is accepted. Remaining ambiguous matches are eliminated by accepting the minimum residual match for each left contour, and then for each right contour.

STEP 7 If two connected contours in the left image match two connected contours in the right, join them and recompute the disparity functional. If the new residual is much larger, reject the join.

We illustrate the results of this algorithm with two synthetic stereo pairs. The images are binary, 256×256 pixels in resolution and were generated with the projection equations (1a) and (1b) with a stereo baseline B of 5 and a field of view of 20 degrees. The first stereo pair (Figure 3a) views a planar surface $Z = 100 + X + Y$, the second (Figure 4a) an elliptic paraboloid $Z = 100 + X + Y + 0.0175X^2 + 0.0175Y^2$. The contours were generated by orthographically projecting ellipses onto the surfaces. There are about 35 contours in each image.



a) stereo pair

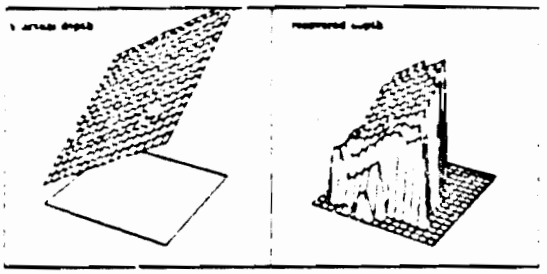
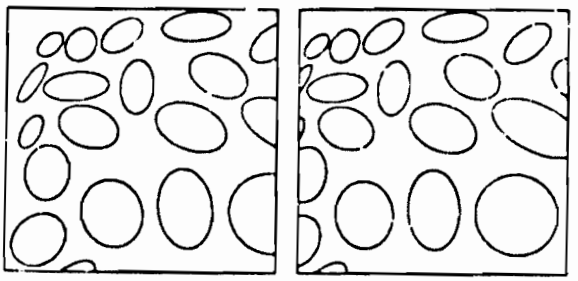


Fig. 3 - Contours on a Plane



a) stereo pair

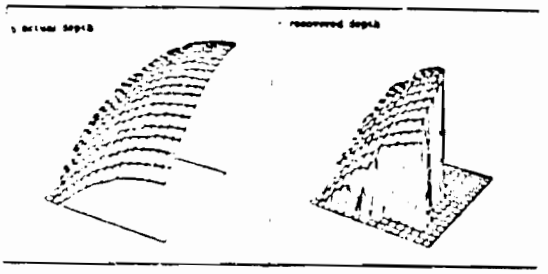


Fig. 4 - Contours on an Elliptic Paraboloid

When run on these images, the contour algorithm was successful in matching all contours except for a few that subtend small fields of view. This is the major drawback of this approach, for if the image texture is too fine to yield contours of adequate field of view, then the recovered disparity gradients will not be reliable. In this case contours need to be grouped together until the field of view is adequate. This is implicit in the neighborhood algorithm presented next. Figures 3b and 4b show, in world coordinates, the original surfaces, while Figures 3c and 4c show the recovered surfaces. A second order disparity functional was used during surface reconstruction in Figure 4c.

3.2 Neighborhood-based algorithm

The neighborhood algorithm avoids the need to search for contour groupings by fitting the disparity functional to a set of neighborhoods in the left image. Each neighborhood searches for the disparity functional that best fits a subset of the local zero crossing matches at two scales. The search process for the best fit is brute force; a neighborhood in the left image is positioned over the right image at each possible disparity offset, and those matches which are closest to this constant disparity, and fall within a matching window, are accepted. These matches are fit with a disparity functional, and the average residuals computed. The process of setting matches and computing the functional is then iterated; outlying matches which differ significantly from the functional approximation are rejected, new matches which now fall within the window are accepted, and the functional is recomputed on the new set. This is first done at the low resolution and then repeated at the high resolution, with a significant difference; at each disparity offset, the high resolution begins with the last disparity functional computed at the low resolution. The disparity offset with the lowest high resolution residual is accepted as correct, and the disparity functional computed at that offset is used in reconstruction. The following steps are performed for each neighborhood.

STEP 1: The low resolution left and right images are scanned to compile a list of potential matches for each edge point in the left image. The restrictions are positive disparity (i.e., match lies to the left) and same zero crossing sign. The same is done at the high resolution.

STEP 2: The intial disparity offset is initialized to 0, the final to the value that would leave the neighborhood 90% off the left side of the right image. The next steps are performed for each offset between the intial and final values.

STEP 2a: For each point in the low resolution left image, the disparity closest to the disparity offset is selected unless it lies outside of a window centered at the disparity offset. The size of the window was $\pm W$, where W was the width of the inner positive region of the zero crossing operator. A linear disparity functional is fit to the matches, and the residual E computed.

STEP 2b: Step 2a is iterated at low resolution, with one difference. The matches selected are now those within $3E$ of the disparity predicted by the disparity functional computed at the last resolution. The iteration proceeds until one of four conditions is satisfied: more than half the points have no match, the residual stops changing, the disparity gradient of the functional exceeds 1, or an arbitrary limit on the number of iterations is reached.

STEP 2c: Step 2a is now applied to the high resolution images with the disparity offset replaced by the low resolution disparity functional. The size of the search window is again $\pm W$, where W was the width of the inner positive region of the high resolution zero-crossing operator.

STEP 2d: Step 2b is applied to the high resolution images.

STEP 3: The disparity offset with the minimum high resolution disparity functional is accepted (provided that at least 50% of the points had matches.) Both the high resolution disparity functional computed at this offset, and the matches that agree, are accepted.

This algorithm has been applied to the three synthetic stereo pairs in Figures 5, 6 and 7. The images are 320×320 pixels in resolution and were generated with the projection equations (1a) and (1b) with a stereo baseline B of 5 and a field of view of 25 degrees. The first stereo pair (Figure 5a) views a planar surface $Z = 100 + X + Y$, the second (Figure 6a) an elliptic paraboloid $Z = 100 + 0.05X^2 + 0.05Y^2$, the third (Figure 7a) a hyperbolic paraboloid $Z = 100 + 0.05X^2 - 0.05Y^2$. The same random dot texture was orthographically projected onto each surface. Zero-crossings of a $\nabla^2 G$ operator were found at two scales with inner windows of 18 and 9 pixels, respectively. The zero-crossing contours shown are not closed because horizontal runs were approximated by the pixel closest to the midpoint. Two neighborhood sizes were used: 64×64 (5×5 degrees) and 32×32 (2.5×2.5 degrees). Each neighborhood size was considered at two resolutions. Only the 256×256 central areas of the left images were matched. The match windows used in steps 2a and 2c were ±18 and ±9 pixels respectively, based on the Marr and Poggio [8] analysis of zero-crossing densities. (This implies that the maximum disparity change across the 64×64 window in step 2a is 36 pixels, giving a maximum disparity gradient less than 1.) A limit of three iterations was set for combined totals of steps 2a and 2b, and 2c and 2d.

The original and recovered surfaces are displayed in Figures 5d-g, 6d-g, and 7d-g. If we consider as correct those zero-crossing matches that are in close agreement with correct disparity functionals, about 90% of the zero-crossings in each image were matched. Despite the small scale roughness of the images, the surfaces were recovered well in all but a few 32×32 neighborhoods. The correct disparity offset was found for these cases, but the disparity functional did not approximate the surface well. This seems due to two effects, both traceable to one

cause. The failures occured on the left side of the images of planar and elliptical surfaces. The part of the surface visible at the left edge of the left image is cut off on the right. This distorts the zero-crossings slightly and eliminates some matches so the horizontal field of view is shortened. The problem may also be due to the steeper slope on the side of the elliptic surface, since the search strategy clearly favors low values for the disparity gradient. There were two substantial failures out of 196 32×32 neighborhoods.

The graphs in Figures 8 and 9 plot the percentage of zero-crossings uniquely matched and the residual, respectively, as a function of disparity offset. The residual is given in units of pixels. (Uniquely matched means there was only one possible zero-crossing match in the search window.) There are six graphs in Figures 8 and 9, one for each of the iterations at low and high resolution. They are for the upper right 64×64 neighborhood in the planar stereo pair of Figure 5. The parameters used in generating the surface were $Z_0 = 100$, $P = 1.0$ and $Q = 1.0$, which give theoretical disparity functional coefficients of $a = 0.050$, $b = -0.050$, $c = -0.050$. The disparity offset selected by the algorithm for this neighborhood was -22, the coefficients of the recovered disparity functional were $a = 0.0501$, $b = -0.0484$, $c = -0.0479$, the residual 0.42 pixels and the recovered depth parameters $Z_0 = 99.7$, $P = 0.97$ and $Q = 0.95$. There were 282 and 511 zero-crossings in this neighborhood at the low and high resolutions, respectively, of which 275 and 469 were given matches. The plot of match percentage for the first low resolution iteration (Figure 8a) shows a variation on the local support measure of Marr and Poggio [8]; probabilistic arguments based on the density of zero-crossings justify rejecting a disparity offset if less than 70% of the zero-crossings in a neighborhood are uniquely matched, i.e., only one potential match appeared within a search window on the scale of the zero-crossing operator. Figure 8a clearly shows the Marr-Poggio constraint, since close to the correct disparity offset the match percentage is 80% but elsewhere it generally stays below 70%. The graphs of the residual in Figure 9 show a distinctive minimum in the area of the correct disparity offset. The residual at the third high resolution iteration (Figure 9f) is the measure used by the algorithm to select the minimum residual for a disparity offset, but the other five iterations show a minimum in the same location. Actually, iterating at the high resolution had little effect on the residual. Away from the correct offset, the residual hovered around the space constant σ of the zero-crossing operator.

The algorithms and experiments presented here have several shortcomings, and mainly serve to illustrate the disparity functional residual as a measure of local support. The neighborhood algorithm is slow (on the order of two CPU hours when implemented in C on DEC VAX 11/785). But, this time resulted from running all six iterations at every disparity offset to collect complete data for illustrative purposes. It is unnecessary to sample

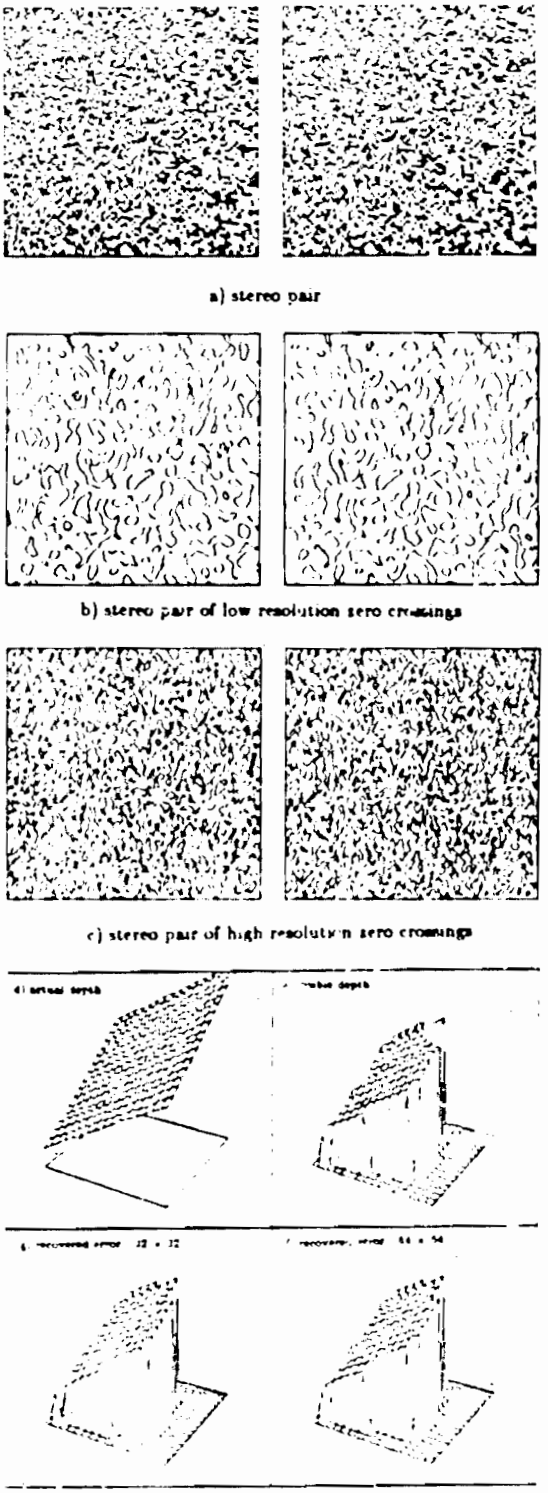


Fig. 5 - Random Texture on a Plane

disparity offsets closer than about half the mask size used in the contour extraction. In addition, the iterations at high resolution are unnecessary. We expect a reduction in computation by a factor of ten. Also, different neighborhoods can be processed simultaneously on a parallel machine. In future work, we expect to refine the algorithms and subject them to more exhaustive testing on natural scenes. The major issues to be investigated are how best to recover the deformation parameters in more complex scenes with transparency and occlusions, and how to segment the resulting disparity field. These issues are discussed in greater detail in Eastman and Waxman [3].

REFERENCES

[1] R.D. Arnold and T.O. Binford, "Geometric Constraints in Stereo Vision", *Proceedings SPIE*, Vol. 238, pp. 281-292, San Diego, CA, 1980.

[2] S. Castan and J. Shen, "A Stereo Vision Algorithm taking into account the perspective distortions", *Proceedings, Seventh International Conference on Pattern Recognition*, pp. 441-443, 1984.

[3] R.D. Eastman and A.M. Waxman, "Using Disparity Functionals for Stereo Correspondence and Surface Reconstruction", University of Maryland, Center for Automation Research Technical Report 145, October, 1985.

[4] W.E.L. Grimson, **From Images to Surfaces** (Cambridge: MIT Press), 1981.

[5] K. Kanatani, "Structure from Motion without Correspondence: General Principle", *Proceedings, Ninth International Joint Conference on Artifical Intelligence*, pp. 886-888, 1985.

[6] J.J. Koenderink and A.J. van Doorn, "Geometry of Binocular Vision and a Model for Stereopsis", *Biological Cybernetics*, Vol. 21, pp. 29-35, 1976.

[7] H.C. Longuet-Higgins and K. Prazdny, "The Interpretation of a Moving Retinal Image", *Proceedings Royal Society London*, Vol. B 208, pp. 385-397, 1980.

[8] D. Marr and T. Poggio, "A Theory of Human Stereo Vision", *Proceedings Royal Society London*, Vol. B 204, pp. 301-328, 1979.

[9] J.E.W. Mayhew and J.P. Frisby, "Computational and Psychological Studies Towards a Theory of Human Stereopsis", *Artifical Intelligence*, Vol. 17, pp. 349-385, 1981.

[10] J.E.W. Mayhew and H.C. Longuet-Higgins, "A Computational Model of Binocular Depth Perception", *Nature*, Vol. 297, pp. 376-378, 1982.

[11] S.B. Pollard, J.E.W. Mayhew and J.P. Frisby, "Disparity Gradients and Stereo Correspondences", Technical Report, Sheffield University, 1985.

[12] K. Prazdny, "Detection of Binocular Disparities", *Biological Cybernetics*, Vol. 52, pp. 93-99, 1985.

[13] L. Quam, "Hierarchical Warp Stereo", *Proceedings, Image Understanding Workshop*, pp. 149-155, 1984.

[14] M. Suk and H. Kang, "New Measures of Similarity between Two Contours Based on Optimal Bivariate Transforms", *Computer Vision, Graphics and Image Processing*, Vol. 28, pp. 168-182, 1984.

[15] D. Terzopoulos, "Multiresolution Computational Processes for Visual Surface Reconstruction", *Computer Vision, Graphics and Image Processing*, Vol. 24, pp. 52-96, 1983.

[16] C.W. Tyler and E.E. Sutton, "Depth from Spatial Frequency Difference: an Old Kind of Stereopsis?", *Vision Research*, Vol. 19, pp. 859-865, 1979.

[17] A.M. Waxman, "An Image Flow Paradigm", *Proceedings 2nd IEEE Workshop on Computer Vision: Representation and Control*, Annapolis, MD, 1984.

[18] A.M. Waxman and J.H. Duncan, "Binocular Image Flows: Steps toward Stereo-Motion Fusion", University of Maryland, Center for Automation Research Technical Report 119, May, 1985.

[19] A.M. Waxman and S. Ullman, "Surface Structure and 3-D Motion From Image Flow: A Kinematic Analysis", University of Maryland, Center for Automation Research Technical Report 24, October 1983.

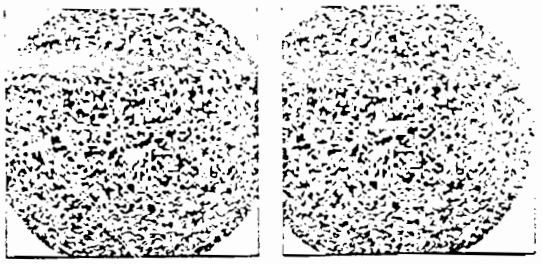
[20] A.M. Waxman and K. Wohn, "Contour Evolution, Neighborhood Deformation and Global Image Flow: Planar Surfaces in Motion", University of Maryland, Center for Automation Research Tech Report 58, April, 1984. Also see *International Journal of Robotics Research*, Vol. 4, 1985.

[21] K. Wohn, "A Contour-Based Approach to Image Flow", Ph.D. Dissertation, Department of Computer Science, University of Maryland, 1984.

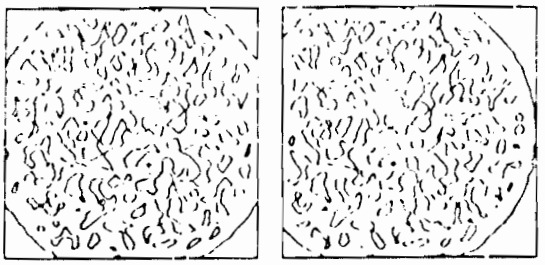
[22] K. Wohn and A.M. Waxman, "Contour Evolution, Neighborhood Deformation and Local Image Flow: Curved Surfaces in Motion", University of Maryland, Center for Automation Research Technical Report 134, July, 1985.

[23] K. Wohn and A.M. Waxman, "The Analytic Structure of Image Flows: Deformation and Segmentation", University of Maryland, Center for Automation Research Technical Report, (in preparation).

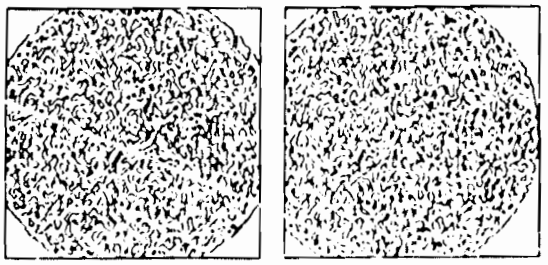
[24] A.L. Yuille and T. Poggio, "A Generalized Ordering Constraint for Stereo Correspondence", M.I.T. A.I. Memo 777, May, 1984.



a) stereo pair



b) stereo pair of low resolution zero crossings



c) stereo pair of high resolution zero crossings

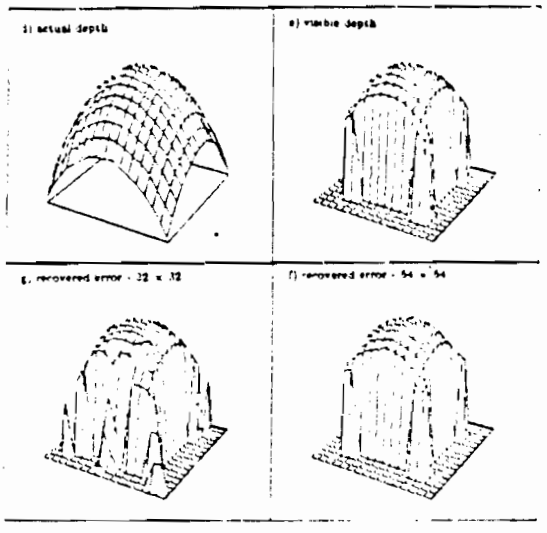
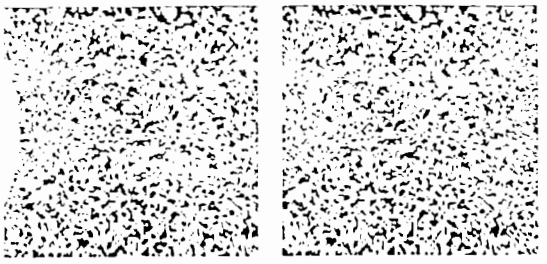
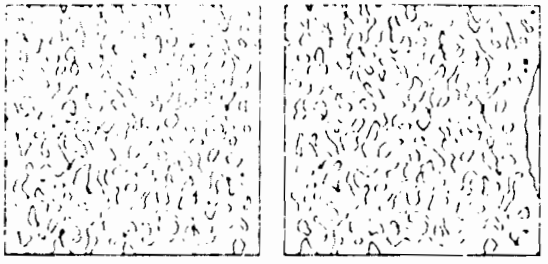


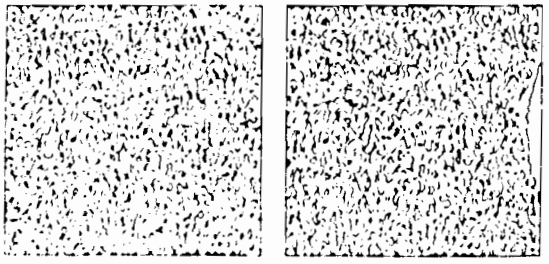
Fig. 6 - Random Texture on a Elliptic Paraboloid



a) stereo pair



b) stereo pair of low resolution zero crossings



c) stereo pair of high resolution zero crossings

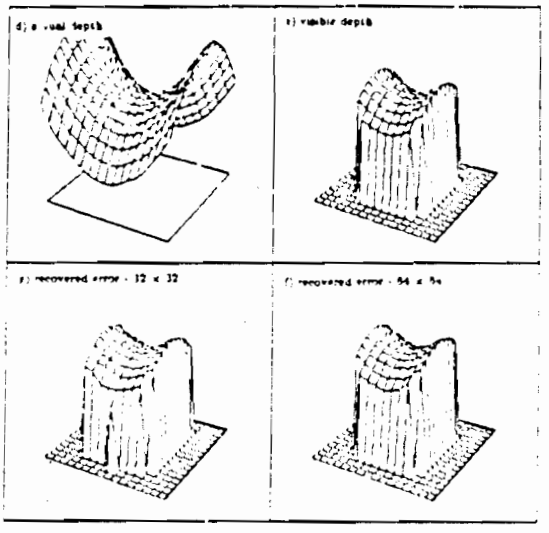
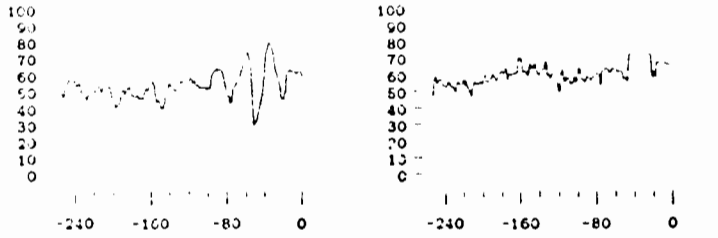
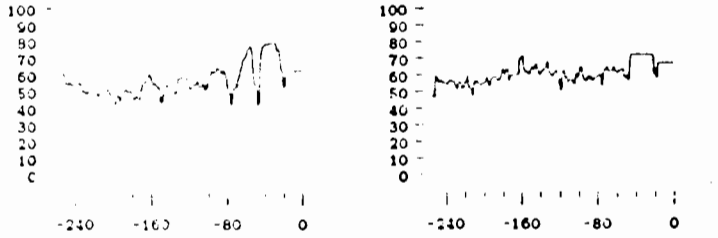


Fig. 7 - Random Texture on a Hyperbolic Paraboloid



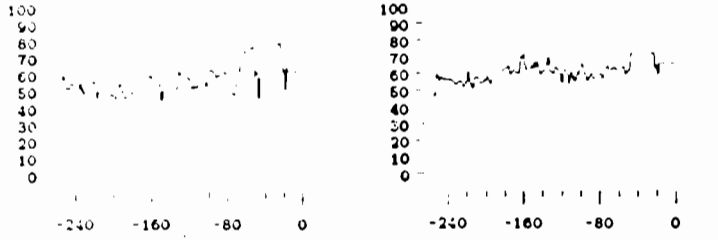
a) First iteration, low resolution

d) First iteration, high resolution



b) Second iteration, low resolution

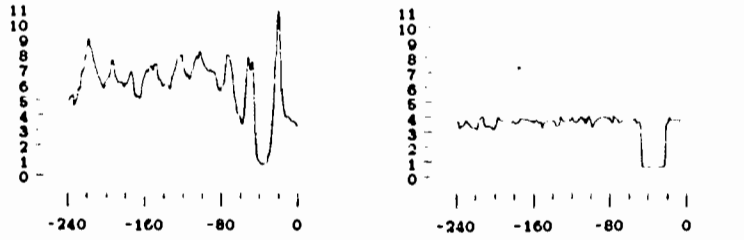
e) Second iteration, high resolution



c) Third iteration, low resolution

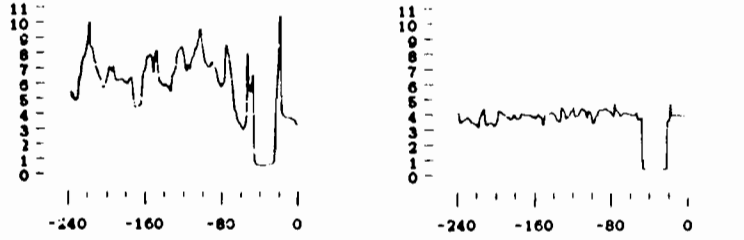
f) Third iteration, high resolution

Fig. 8 -Percentage uniquely matched features for single neighborhood
Vertical axis - percentage of points with match
Horizontal axis - disparity offset in p



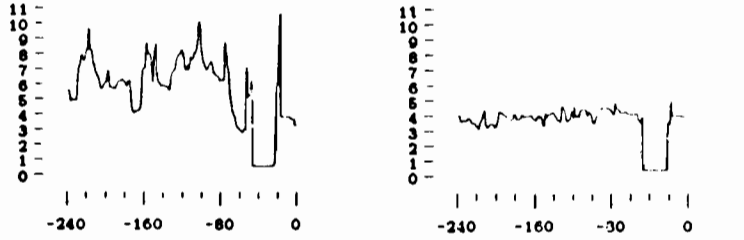
a) First iteration, low resolution

d) First iteration, high resolution



b) Second iteration, low resolution

e) Second iteration, high resolution



c) Third iteration, low resolution

f) Third iteration, high resolution

Fig. 9 - Residual for single neighborhood.
Vertical axis - mean absolute disparity error in pixels.
Horizontal axis - disparity offset in pixels.

GC970145-31

Evidence Combination for Vision using Likelihood Generators

David Sher
Computer Science Department
University of Rochester
Rochester NY 14627
November 7, 1985

1. Objectives

My project is about building image understanding systems. In particular, I am studying how to design a system that takes a color image from a camera and finds the outlines of the objects in the image. A representation of outlines is often called a segmentation of the image. The task of generating a segmentation is called image segmentation.

Outlines are extremely useful for other low-level vision tasks. The algorithms that benefit most from knowing outlines are algorithms that are based on a local continuity or planarity assumption. The standard relaxation based shape from shading [12] and optical flow [4] [5] algorithms are examples of such algorithms. Intermediate level vision algorithms have been developed to take outlined regions and associate them with known objects [11] [15]. Thus the task of image segmentation can be considered important, even fundamental, to computer vision. Few algorithms (other than segmentation algorithms of course) can be run effectively on an unsegmented image.

The objective in this project is to build programs that can take expert knowledge about the appearance of outlines in images and derive a segmentation. Any knowledge about the structure of images should be representable in my framework. Using the expert knowledge, the system should return the most useful information about the outlines of the objects in the image possible.

Much of the work done in computer vision has been developed with different goals in mind. Because of the difference in goals the algorithms some people developed have serious shortcomings from my viewpoint. One alternate set of objectives is those held by researchers inspired by biological modeling. An excellent work in biological modeling is that of Fleet [9]. His work is on the temporal and spatial characteristics of center surrounds operators.

When working on modeling one tries to develop algorithms whose behavior closely approximates that of a human vision system. An example of such approximation is to have only band limited operators because the cells on the mammalian optic nerve have been shown to be band limited. For my work this limitation is not sufficient reason to use exclusively band limited operators. If it is shown that the phenomena that I am trying to detect are band limited or that a band limited operator is sufficient to detect the phenomena without loss of accuracy then I would use band limited operators.

I claim that one criterion in most biological modeling studies is to come up with simple and easily analyzed operators. Nonlinearities in operators must be justified by showing some nonlinearity in the behavior of the biological system that can not be modeled with linear operators. For my work, I have to justify linearity in an operator by demonstrating some linearity in the world.

Another viewpoint of workers in computer vision is derived from working on signal processing. Signal processing shares many of the objectives of computer vision. Signal processing developed as a discipline on the development of radar and sonar. Signal processing predated electronic computation as we know it today. Thus much of the work in signal processing is on the behavior of detectors that can be made from simple electronic circuitry. Many *optimality* results in signal processing are only for linear operators.

Signal analysis originated with analysis of a single time varying signal and many of its results apply only to such a model. Such models assume that images have properties common to time varying signals that images don't share like causality. When a signal has a causal model (in work in signal analysis) then the best prediction of the value of a signal at a point can be made without knowing the behavior of the signal in the future only the past behavior. However, it is not true that the left side of the image predicts the values of the middle ignoring the right side. Despite this fact, image processing operators have been recently described that assume such causality [8].

2. Expertise

My project studies image interpretation. An image interpretation system takes assumptions about the structure of the scene being observed and uses them in combination with the image to generate an interpretation. The original source of assumptions is the human programmer. My project is about taking information from human experts on image processing and deriving a system that uses that information for image interpretation.

There are a variety of sources of expertise in image processing. There are more sources of expertise at the low level (processing done before segmentation) than at the higher levels (processing that assumes segmentation) which is why my project concentrates on the low level systems. Some knowledge is about typical arrangements of objects in the world and typical colorations. Other knowledge is about the physics of observation. Such knowledge can be used to build boundary detectors.

Other sources of knowledge are the operators designed to detect various features of images. I intend to incorporate the knowledge that went into the construction of such operators into my system. A problem with such operators is that the assumptions that were used to build them are often implicit and unrecognized even by the author of the operator. Thus part of my project is devoted to recovering the assumptions behind known operators. Such information is useful outside the context of my project.

To understand how to turn knowledge into operators or analyze operators for the knowledge within them, it helps to know certain facts about expert knowledge on image understanding. Expert knowledge is compartmentalized. Knowledge exists about how to use shading to derive surface orientation [12]. Knowledge exists about how to derive surface orientation from texture in the image [14] [13] [1]. But little work has been done to derive surface orientation from both shading and texture even though both exist in most images. Thus an image interpretation system has to derive information from a variety of knowledge sources whose expertise is limited to a particular set of situations that use part of the information in the image.

The image interpretation system that I am building contains a set of boundary detectors (developed using different sources of expertise). I have developed techniques for integrating sets of detectors into a system for detecting features such as boundaries.

The relationship between an image and the imaged scene is not one to one. There are many scenes that can cause any particular image. Thus images are inherently ambiguous. Ambiguity makes it unreasonable to expect to derive a boundary detector that gives as output a decision of whether a boundary exists or not at a point. A probability of the boundary existing is a more reasonable output. Thus the detectors that are used by the image interpretation system in my project have probabilities as output.

3. Theories

Two theories are required for my project. (Theory is used here in the sense of a body of knowledge and techniques such as quantum theory.) I need ways to take human expert knowledge on a specific subject, such as the probability of boundaries given particular texture edges, and generate an algorithm that calculates the probability of some feature of an image such as the probability of an object boundary in a part of that particular image (feature will be precisely defined in the further sections). Such a theory is a theory of feature detection. The other theory is about combining the information generated by a set of feature detectors for the same feature. The detectors are based on different assumptions or take into account different data. My evidence theory is about combining the output of a set of detectors.

4. Definitions of Fundamental Terms

The *conceptual universe* contains not only all observations but all structures used by various models (or domains as explained later) to explain and model the observations. One such structure is an array of boundary features (feature is defined in the next paragraph) that are 1 when a boundary passes through their window in the image and 0 otherwise. In this work I take the reasonable philosophical position that there is a real world being imaged. The conceptual universe is a representation of relevant aspects of this world and of the image.

A *feature* of the conceptual universe is either a particular observation or a parameter of a model (domain) used to understand the observed image. Features take on values and a feature's value is useful in understanding the scene being observed. Note that in signal analysis, biological modeling, and computer vision feature is used in a more restrictive sense, to refer to the output of functions applied to the observed image. While my definition of feature does not conflict with this usage it is an extension. A *feature space* is a set of values that a feature of the conceptual universe can attain.

An example of a feature is a boundary feature. A *boundary feature* takes on the value boundary when there is a region boundary passing through the part of the image assigned to it. Otherwise it has the value no boundary. A boundary feature's feature space is the set (boundary, no boundary). Another kind of feature is a *surface orientation feature* that takes on values according to the slant and tilt of the surface of the object imaged. Its feature space is a set of ordered pairs of numbers. Another kind of feature is the intensity observed at a point in the image, an *observed intensity feature.*

Features are generally associated with points in either the image or the three dimensional scene. In my work I concentrate on features associated with the two dimensional image.

Every image representation in a computer must consist of a finite set of numbers (since an infinity of numbers is too expensive to store). Thus every representation of the conceptual universe in a computer must be discrete. However the underlying model of the scene could be continuous and much vision and image reconstruction work is done on discrete approximations of continuous models [16] [3].

My entire conceptual universe including the models is discrete. Thus everything in my system can be directly represented in my system. This eases evidence combination.

5. Domains

In a previous section I described certain properties of expert knowledge. A domain is a formal device for manipulating expert knowledge. A *domain* is a set of axioms (logical statements) that taken in conjunction describe a body of expert knowledge. A domain can be considered a logical statement and manipulated the same way. A *situation* is an assignment of values to features of the conceptual universe. A domain is equivalent to the set of situations that do not contradict the axioms in the domain. Thus a domain can be manipulated in the same way as a set.

An example of a domain (called D_1) is the set of statements below.

(1) The image is of a set of regions of uniform reflectance.

(2) Each region has a finite extent with a well defined boundary.

(3) Each boundary consists of a finite set of corners with curves of low curvature between them.

(4) Each surface has slowly varying surface orientation.

(5) All surfaces are illuminated from the same direction with uniform intensity.

The domain above is realistic but rather complex. A simpler domain has been used in my work so far and described later on.

A subset of D_1 is a domain, D_2, that assumes the same as above and that all the surfaces are perpendicular to the line of sight. D_2 applies to a subset of the situations D_1 applies to. If the axioms of D_1 are taken in conjunction with the axioms of D_2 and the result is D_3.

6. Markov Random Fields

In the domains I am using I assume that the probability of feature values can be determined exactly if I know the values of a set of nearby features. One such assumption can be described thus. A boundary feature is associated with a window in the image. It attains the value boundary when a boundary passes through that window. Otherwise it attains the value no boundary. If I know the values of the pixels of the window associated with a boundary feature, $W(f)$, and the values of the boundary features with areas around $W(f)$ then I could determine the probability of that boundary feature taking on the value boundary independent of the values attained by the rest of the features in the conceptual universe. Such an assumption of locality is made in most of the work in vision and signal processing. For each feature there is a set of features that determine its probability distribution. This set defines a function, N, where F is the set of all features in the conceptual universe:

$$N: F \rightarrow 2^F$$

The combination of F and N describes a neighborhood system on F. A set with a neighborhood system is a *lattice*. A *setting* of a subset, S, of F (such as $N(f) \in D$) is an assignment of values, $v \in V$, to the members of F. It is a function:

$$a: S \rightarrow V$$

A setting can also be described as an assignment of values to S from V. It also can be considered a set of ordered pairs: $(s \in S, v)$ where every member of S appears on the left once and only once.

The domains I use describe a function p_f that takes a setting $a \in A_{S(f)}$ on $S(f)$ and returns a probability distribution on the feature space, V_f of f. In mathematical notation:

$$p_f: A_{S(f)} V_f \rightarrow (0,1)$$

The set of p_f combined with the lattice described by N and F describe a *Markov random field*. The domains I am using imply Markov random fields. (Actually any

domain sufficient to specify p_r is trivially a Markov random field where $N(r)=R$ but in the cases I am interested $N(r)$ is small relative to R.)

Markov random fields have been used as a model for a variety of image restoration algorithms [10] [7] and for recognizing texture [6].

7. Hidden and Observed Features

An image analysis system often has the goal of deriving implicit or "hidden" scene parameters from the observed data. There may be other hidden parameters that are not wanted but are useful as intermediate results. Assume an imaging system needs to determine where the lines are in an observed intensity image. The lines are the desired hidden parameter while the intensity image is the observed data. Often the edges in the image are useful in determining where the lines are. The edge configuration is also a hidden parameter. The models I am using for the data consist of a network of features. Some of them take on values according to observed data. Such are *observed features*. The other features represent structures in the image that are not directly observable. Such are *hidden features*.

8. Feature Detectors

Most human expertise about low level image interpretation can be described in terms of feature detection. A *feature detector* is an algorithm that outputs information about the value of a particular feature. Such information translates to (or is) a probability distribution over the feature space of the feature. As an example a boundary detector returns the probability that a boundary passes through an area. For each point in the image there is a feature that represents the existence or nonexistence of a boundary at a point. If the boundary detector returns p for an area then the corresponding boundary feature has a distribution of $(p,1-p)$ on the values (boundary,no boundary). A feature detector takes as input the values of the observed features and, possibly, estimates of the distributions on neighboring features. Most established feature detection algorithms can be described in such a manner.

Note that from a lattice of hidden and observed features, and from a feature detector, a domain can be derived. That domain simply states that the probability distribution of a feature, given a setting of its neighborhood, is what the feature detector says it is. A useful domain implies a feature detector also.

9. Likelihood Generators

Often it is easier to state and solve the inverse vision problem (which is why computer graphics can generate realistic images that current image understanding systems can't analyze). It may be easier to describe the probable structure of an observed intensity image in the presence of a boundary than to describe the probability distribution on the boundary given an observed image. The probability that the observed features are assigned according to a setting a when a hidden feature, f, takes on value v is the *likelihood* of v for a. I use as short hand notation for this $l_f(a|v)$. A *likelihood generator* is an algorithm that uses a domain D to estimate the likelihood of v for a. Thus I use as notation for the output of a likelihood generator as $l_f(a|v\&D)$. Given a likelihood generator for D and a prior estimate of the distribution of f's values then one can make a feature detector for f using Bayes' Rule:

$$P_f(v|a\&D)=\frac{l_f(a|v\&D)prior_f(v)}{\sum_{v'\in V} l_f(a|v'\&D)prior_f(v')} \qquad (1)$$

I call the feature detector thus derived a *Bayesian feature detector* for domain D.

The set of likelihoods for a feature f given an observation a contains more information than (1) uses. The denominator in (1)

$$\sum_{v'\in V} l_f(a|v'\&D)prior_f(v') \qquad (2)$$

is the probability that a would occur given the prior estimate of the distribution on f's feature space. If the probability is too low then the domain being used probably is not correct. I use this information combined with *a priori* information about the reliability of the domain to derive an evidence theory further on.

10. Evidence Theory

The fundamental question I address is how to combine two feature detectors. Each feature detector implies a domain. To derive a combined feature detector, I need to show how to combine two likelihood generators given some prior information on the two domains and their intersection. If I can combine the output of two likelihood generators I can use the combined generator to make a feature detector using Bayes' rule. Then I have a feature detector that returns useful data when either of the two domains apply. In this section I use O_f to represent a setting of the features other than f.

10.1. Combining Likelihoods

If I have a likelihood generators for two domains, D_1 and D_2, I would find useful a likelihood generator that works for either domain. I call the domain that holds when either D_1 or D_2 holds $D_1 \vee D_2$. In this section I show how to make such a likelihood generator.

I assume that I can derive or know certain information *a priori*. The *a priori* information is the probabilities of D_1, D_2, and $D_1 \& D_2$ holding. One source of *a priori* knowledge is statistics acquired by human interpretation of a test suite of images. Other sources could be a model of the scenes expected to be observed by the imaging system.

I also assume that the *a priori* probability of the feature taking on any value is the same under any of my domains or combinations thereof. Thus all the domains have the same prior bias regarding the feature. Given such assumptions, I can derive a combination rule for likelihoods:

$$L(O_f = v \,|\, O_f \& D_1 \vee D_2) = \frac{\begin{array}{l} L(O_f = v \,|\, O_f \& D_1) P(D_1) \\ + \\ L(O_f = v \,|\, O_f \& D_2) P(D_2) \end{array}}{P(D_1) + P(D_2) - P(D_1 \& D_2)} \quad (3)$$

$$L(O_f = v \,|\, O_f \& D_1 \& D_2) = \frac{L(O_f = v \,|\, O_f \& D_1 \& D_2) P(D_1 \& D_2)}{P(D_1) + P(D_2) - P(D_1 \& D_2)}$$

O_f is an assignment, $O_f : f' \neq f \rightarrow V$, of all the features besides f.

Equation (3) contains a new term $L(O_f = v \,|\, O_f \& D_1 \& D_2)$ which is the likelihood generator whose domain is the conjunction of the axioms of D_1 and D_2. In the special case where the two domains are disjoint ($P(D_1 \& D_2) = 0$), $L(O_f = v \,|\, O_f \& D_1 \& D_2)$ is multiplied by 0 so is irrelevant. Here, the output of the combined likelihood generators is the weighted average of the outputs of the two other generators.

An example of two disjoint domains is one that assumes that objects in the are world are Lambertian surfaces, and another that assumes that the objects have specular reflectance properties. Both domains can not be simultaneously true at any point in the image (they both can be simultaneously false though).

Another example of two disjoint domains are one domain that assumes gaussian additive noise of standard deviation $4 \pm \varepsilon$ and another that assumes gaussian additive noise of standard deviation of $8 \pm \varepsilon$. Both domains can not hold simultaneously. I use these domains in experiments that test the effectiveness and properties of my evidence combination rules on real images (see further on).

If one domain is a subset of the other then the output is that of the superset since the feature detector for the superset is presumed to have already taken the subset into account. If D_1 and D_2 have have the independence property described by equation (4) then the output is bilinear in the output of the two feature detectors.

$$L(O_f = v \,|\, O_f \& D_1 \& D_2) = \gamma(D_1, D_2) L(O_f = v \,|\, O_f \& D_1) L(O_f = v \,|\, O_f \& D_2) \quad (4)$$

Conditional independence satisfies (4) with $\gamma(D_1,D_2)$ always equal to 1. Bilinearity is a somewhat more flexible criterion than that of being disjoint since $\gamma(D_1,D_2)=0$ means that the two domains are disjoint. When the domains are not disjoint or bilinear the situation is dealt with as a special case. (Perhaps techniques for dealing with cases where the conjunction of domains is significant will be developed during my research.)

Note that the contribution of each likelihood generator is proportional to the *a priori* probability of its domain holding. The contribution of the likelihood generator is also proportional to the value in equation (2) which corresponds to the reliability of the operator. Thus likelihood generators use the information in the image about their own reliability when being combined.

10.2. Fred and the Thermometer

To illustrate my evidence theory I would like to introduce a simple case originated by Glenn Shafer. It is about detecting the temperature outside, given two sources of expertise. There is a thermometer visible outside. Fred has just walked into the room, here in Rochester NY. Fred is an academic type. About 20% of the time he is in his own world and thus his statements have no understandable connection to the outside world. When it is freezing he usually doesn't comment. When it isn't he comments on it about 50% of the time. The thermometer is a standard outside thermometer and thus subject to break down. There is about a 5% chance that the thermometer is broken. The question is whether it is freezing outside. Fred says, "Its nice and warm outside." The thermometer reports 29 degrees Fahrenheit (approximately -2 C). On days this time of year (fall) theres a 50-50 chance of freezing weather. Table 1 shows the likelihoods generated by Fred and the thermometer.

Table 1 : Likelihoods for Freezing Weather

Event	Fred's	Thermometer's
Freezing Weather	0.00	1.00
Warm Weather	0.50	0.00

There are two domains, the Fred domain and the Therm domain. The information from the domains about the weather is shown in Table 1. Note that there is no information about what happens when all of my information sources are not reliable. Currently my theory assumes that this never happens. I adjust my evidence theory to deal with the possibility of total unreliability in a later section.

Because I ignore the possibility that everyone is unreliable if only Fred's evidence is available then I am 100% sure of warm weather. If only the thermometer is seen I am 100% sure of freezing weather. Thus I have two high confidence evidence sources conflicting.

I believe these things about the relationship between the two domains:

(1) The Fred domain is 80% reliable.

(2) The Therm domain is 95% reliable.

(3) The reliability of the Fred domain is independent of the Therm domain. Thus both domains are reliable 76% (95% of 80%) of the time.

(4) The intersection between the Fred domain and the Therm domain is 0 when we make the observations we have.

This information about my domains is sufficient for me to apply the likelihood combination rule.

I can combine the likelihoods from my two domains using the likelihood combination rule derived in the preceding section (Equation (3)). I need to handle the case when both Fred and the thermometer are in working order. The statements of Fred and the thermometer contradict. But these statements are impossible if Fred and the thermometer are giving accurate readings. So the operator when Fred and the thermometer are operational returns 0.

Thus by equation (3) the likelihood of freezing weather is the weighted sum of the likelihoods output by Fred and the thermometer weighted by the probability that they are accurate and the result added by their probability of their accuracy minus the probability that both are accurate. Thus the likelihood of freezing weather is:

$$\frac{0.00*0.80+1.00*0.95-0.00*0.76}{0.80+0.95-0.76}=0.96$$

Thus the likelihood of freezing weather is 0.96 . Equation (3) applied to warm weather looks like:

$$\frac{0.50*0.80+0.00*0.95-0.00*0.76}{0.80+0.95-0.76}=0.40$$

Thus the likelihood of warm weather is 0.40 .

If I now apply Bayes rule to these probability with my 50-50 prior I get a 0.70 probability that the weather is freezing.

10.3. Taking Uncertainty into Account

In the example of "Fred and the Thermometer," I calculated the likelihoods and conditional probabilities when at least one domain has its assumptions met. I have up to now ignored the possibility that none of my domains may apply. In this section I show how to take this possibility into account in my system.

I represent the possibility of none of my domains holding by another domain D_i that represents the case where I am totally ignorant of the structure of the universe. Here I use maximum entropy like techniques [Max entropy ref] to make the likelihood generator for this minimalist domain.

For D_i's likelihood generator, $I_i(O_j|\&D_i)$, a good function is the one that preserves the prior. That is the posterior found using a prior is the same as the prior. Such a set of likelihoods act in accordance with a naive conception of ignorance. The generator that preserves the prior assigns equal probability to all O (maximum entropy assumption). The conditional probabilities must sum to 1. Thus the likelihood generator for $I_i(O_j|\&D_i)$ has a well defined value.

10.4. Fred and the Thermometer with Ignorance

Both Fred and the Thermometer were quite reliable. The probability of both of them failing is .01. D_i gives equal likelihood to all observations given either freezing weather or warm weather. I simplify the situation by assuming there are four cases:

(1) Fred says nothing; thermometer reports below 32.

(2) Fred says it's warm; thermometer reports below 32.

(3) Fred says nothing; thermometer reports above 32.

(4) Fred says it's warm; thermometer reports above 32.

D_i gives all 4 cases likelihood .25.

Table 5 describes how the likelihoods change for the case of "Fred and the Thermometer" when the possibility of ignorance is taken into account.

Table 5: Fred and the Thermometer with Ignorance

Event	Ignorance Domain	
	Without	With
Freezing Weather	0.96	0.95
Warm Weather	0.40	0.40

There is not much difference between the two because I have high confidence in both Fred and the thermometer. Using 50-50 as my prior on freezing weather I get .70 as the probability of freezing weather. There is no difference to the nearest .01 in the probability because the ignorance case is unlikely. The domains used for analyzing images are much less reliable so the effect of the uncertainty domain is more pronounced.

For a more dramatic illustration consider the case where I only have Freds input but can not see the thermometer. The result of using D_i here is shown in table 6:

Table 6: Just Fred with Ignorance

Event	Ignorance Domain	
	Without	With
Freezing Weather	0.00	0.10
Warm Weather	0.50	0.50

Without using the ignorance domain I have a probability of 1.00 for warm weather given just Fred's response because I have to ignore the possibility that he is unreliable. With ignorance and a prior of 50-50 I have a probability 0.83 for warm weather and a probability 0.17 for freezing weather.

II. Theory of Feature Detection

The previous section shows that it is simple to develop an evidence theory with likelihood generators. Earlier I showed how to derive a feature detector from a likelihood generator and a set of priors. I called such feature detectors Bayesian feature detectors. This section describes techniques for development of Bayesian feature detectors.

II.1. Developing Feature Detectors Directly from Domains

A benefit of Bayesian feature detector theory is that it is possible to build optimal feature detectors for a domain. The domains I use have axioms that when combined with a prior on the feature space of any feature in my conceptual universe determine a unique posterior on the feature space. A function that calculates precisely the correct posterior probability distribution on the feature space given the observed image and the feature is the optimal Bayesian feature detector for a domain. An example of a domain and its optimal detector is shown further on.

A method for making a Bayesian feature detector starts with a domain for a feature that is sufficiently specific to calculate a probability for any observation, given any of the different values in feature space. Many such domains can be garnered from the image reconstruction literature [2]. Usually in such a domain the feature space values generate with some probability a set of possible ideal images that then are operated on by some noise processes that distort the result. When a domain is of this form, an optimal operator can be derived.

The optimal detector is often too computationally expensive to run on the machine. However perfect accuracy in calculating the probability distribution is often unnecessary (and usually impossible to achieve in any case). If a careful analysis is made of the degree of accuracy required, an algorithm can be derived that calculates, without undue computational expense, a value that is within the required accuracy.

I have built a Bayesian feature detector for a parameterized set of simple domains. The domains can be pictured as gray-level images of shapes (regions) where these assumptions are made:

Region Monochromaticity

> The gray-level of a region does not vary within the region.

Duality

> The shapes are thick enough so that a 3 pixel by 3 pixel window can only fall on two regions. Corners are considered unlikely enough to be ignored. This assumption is normal (sometimes tacit) in line finding work.

Noise

> The scene is viewed through an imaging device that introduces a noise factor into its evaluation of the value of each point according to a known set of conditional probabilities. The noise is a parameter of my implementation.

Histogram

The *a priori* probability of any pixel inside a window being a certain gray level (before noise) is known.

(local) Ergodicity

The imaging device misregisters pixels. If a window falls on a boundary between two regions the result is equivalent to random selection of intensities from the two regions (before noise). If the window lies in the interior of the region, its pixels are all selected from that region (before noise). (To be realistic a more complex assumption is necessary but the ergodicity assumption simplifies computation.)

Window

The likelihood for a boundary feature can be determined entirely from the values of a 3 by 3 window of observed intensities associated with the feature. (This is another assumption that is not realistic but simplifies calculations.)

An optimal Bayesian feature detector for boundaries can be built for this domain by building operators that calculate $I_f(O_f|B)$ and $I_f(O_f|not\ B)$. B represents that there is a boundary at the point represented by the feature (or near it). O_f is the setting of the intensity image. The window assumption says that:

$$I_f(O_f|B) = I_f(W(f)|B)$$
$$I_f(O_f|not\ B) = I_f(W(f)|not\ B) \quad (9)$$

where $W(f)$ is the setting of the window about f. The window and histogram assumptions imply that $I_f(W(f)|not\ B)$ can be calculated by integrating over the different possible gray-levels, the probability of the window occurring, when it is completely within a region of that gray-level. The histogram assumption says that the probability of a window being seen in a region of a particular gray-level can be calculated as the product of the probability of each pixel being seen in a region of that color. The probability of each pixel can be looked up in a list of conditional probabilities according to the noise assumption. To calculate $I_f(W(f)|B)$ it is necessary to integrate over all pairs of gray-levels that two regions can be.

The algorithm implied by the above assumptions is somewhat computationally expensive. In particular iterating through all pairs of colors and doing 9 multiplications and 1 add for each pair results in an algorithm that does nearly 1,000,000 floating point operations *at each point*. I have written a program that uses a modified version of this algorithm on an image to calculate the probability of a boundary at each point. $I_f(W(f)|B)$ was approximated by assuming the pixels resulting from the brighter region all have higher intensity than the pixels resulting from the darker region. Even though a variety of simplifications and approximations were used to generate the feature detector it did a decent job of determining boundaries (comparable to ordinary edge detectors such as Sobel).

I have applied this feature detector using the assumption of uniform probability for regions of all gray levels and gaussian additive noise. I assume a prior of .2 for a boundary passing through any 3x3 window. Figure 1 shows the result of applying my feature detectors to artificial images assuming differing standard deviations for the noise. I use standard deviations of 4, 8, and 16 for testing purposes.

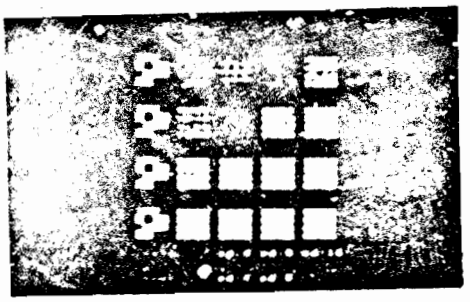


Figure 1:
Applying Approximated Optimal Operator
to Random Checkerboards

Brightness is proportional to the probability of no edge (Black means edge, White means no edge) Left most column is an artificial image corrupted with gaussian additive 0 mean noise of from top to bottom standard deviation 0, 4, 8, and 16 out of 256. The rest of the columns are the output of Bayesian feature detectors. The moving from left to right they are:

The combination (according to the evidence combination rules described previously) with equal weight the three detectors whose output is shown to the right.

(1) The Bayesian feature detector that assumes gaussian additive noise of standard deviation 4 mean 0 out of 256.

(2) The Bayesian feature detector that assumes gaussian additive noise of standard deviation 8 mean 0 out of 256.

(3) The Bayesian feature detector that assumes gaussian additive noise of standard deviation 16 mean 0 out of 256.

Figure 2 shows the result of applying the Sobel edge operator (thresholded) to these artificial images.

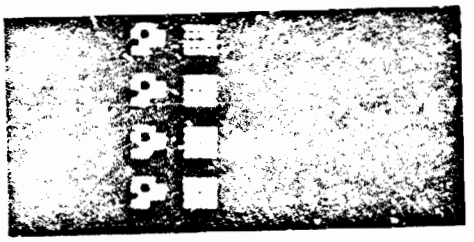


Figure 2:
Applying the Sobel
to Random Checkerboards

Brightness is proportional to the probability of no edge (Black means edge, White means no edge). Left column is an artificial image corrupted with gaussian additive 0 mean noise of from top to bottom standard deviation 0, 4, 8, and 16 out of 256. Right column is the result of running Sobel operator on image normalized into [0,1].

Figure 3 shows the result of applying my feature detectors to a real aerial photograph.

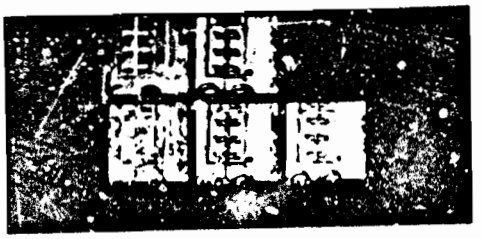


Figure 3:
Applying approximately optimal operators
to aerial photograph

upper left hand picture is the image. Brightness is proportional to the probability of no edge (Black means edge, White means no edge) upper right hand picture is the output of the combined feature detector. lower pictures are the output of the approximately optimal operators for gaussian additive mean 0 noise of standard deviation 4, 8, and 16 out of 256 from left to right respectively.

Figure 4 shows the output of the Sobel.

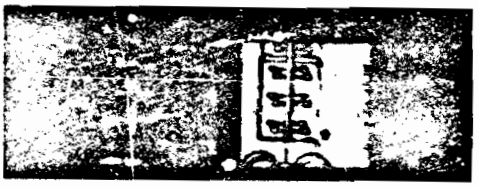


Figure 4:
Applying Sobel
to aerial photograph

left hand picture is the image. Brightness is proportional to the probability of no edge (Black means edge, White means no edge). Right hand picture is the output of the Sobel normalized to [0,1].

I have also used my evidence combination rules to combine several Bayesian features into one that is more flexible. Figure 1 shows the output of the feature detector that represent the equal combination of the three feature detectors on the artificial images. Figure 3 shows the result when the combined feature detector is applied to the aerial photograph.

11.2. Techniques for Deriving Feature Detectors from Domains

I have abstracted four techniques for reducing the computational costs of feature detectors, introducing some inaccuracy in the output of the feature detectors. The techniques are:

(1) Simplifying the Domain:
Changing the assumptions in the domain to make it easier to analyze and build detectors. The ergodicity assumption was a simplification of a more complex assumption that would involve windows with only a corner on a boundary and other such complexities.

(2) Reducing the Scope:
Reducing the data that needs to be examined to evaluate the detectors. The window assumption did that in a explicit way by limiting the amount of the image that needed to be examined to determine the probability of a boundary in the 3 by 3 window to that window. Actually the pixels around the window also might be significant in determining whether a boundary passes through the window. The scoping assumption ignores this possibility. The ergodicity assumption reduced the scope in a more subtle way. As a result of the ergodicity assumption the probability of a boundary can be determined from the histogram of the window rather than the window itself.

(3) Approximating the Feature Detector:
Using standard numerical techniques to approximate the optimal feature detector with functions that are computationally cheaper. As an example, assume the probability of a boundary at a point need only be approximated to the nearest .1, under gauss an additive noise. In a sample image I tested, I determined that more than 90% of the 3 by 3 windows in the sample image had probabilities that are within .1 of 0 or 1 from the range of gray levels in the windows. To determine this I used the fact that when the range of values in a window is larger than a certain value I can prove that the probability it contains a boundary is above 90%. When the range is smaller than another constant then I can prove the probability it contains a boundary is smaller than 10%. Only 10% of the windows in the region have ranges that fall between these two constants.

(4) Finding a Sufficient Statistic:
finding an easy to calculate function or functions of the neighborhood whose output uniquely determines the likelihood. This technique actually loses no accuracy and can be a great computational help.

If the domain is as above and the noise is known to be gaussian additive of known standard deviation then the likelihood of no boundary can be determined from the mean and standard deviation of the window and the likelihood of a boundary can be determined from the means and standard deviations of the pairwise partitions of the windows.

11.3. Analyzing Established Feature Detectors

Another path to building Bayesian feature detectors is to change established feature detection algorithms into Bayesian feature detectors. Many established feature detectors are computationally efficient and extensively analyzed. To change a feature detector into a Bayesian feature detector it is necessary to extract from it the domain it is based on. Then the reliability of the detector can be evaluated and its output combined with the output of other feature detectors.

One way to extract the domain from a feature detector is to consider the relationship between the output of the feature detector and the probability distribution for the feature space. Most feature detectors return output along a linear scale that is supposed to correspond with probabilities. I base my technique for domain extraction on a rigorous version of this correspondence.

I define a feature detector and domain to be *consistent* when according to the domain's assumptions the feature detector is computing a function that is monotonic with the function that is the probability distribution for the feature space according to the domain. To build a domain consistent with an established feature detector I make some assumptions (to simplify the problem) and then attempt to show mathematically what other assumptions are necessary to assure that the domain is consistent with the feature detector.

I have done an analysis of the one-dimensional gradient by finding consistent domains for it. I started by considering only domains with an assumption that the probability of a boundary can be determined from only two points since the gradient only uses two points. I also make the monochromaticity and noise assumptions from the previous section. I also assume that the probability of the gray levels is constant. I found that domains that make these assumptions must have symmetric, unimodal, and additive noise to be consistent with the gradient. I found that domains with a significant occluding noise source (where the noise value replaces the true value rather than adding to it) were not consistent with the gradient.

Consider a domain with gaußian additive noise (a symmetric, unimodal and additive noise). When there is a boundary between them the two pixels in the window are of independent objects. The probability of seeing that window is the product of the probabilities of seeing the gray-levels of each of the two pixels. The probability of seeing any gray-level at a pixel is constant. So the product of the probabilities of the two gray-levels seen is that constant squared which shows $I_f(W(f)|B)$ is a constant.

$I_f(W(f)|not\ B)$ can be calculated from table lookup on the gradient. Thus an efficient Bayesian feature detector can be derived from the I dimensional gradient.

Figures 5, 6 and 7 illustrate the weakness of the gradient given an occluding noise source versus an additive one.

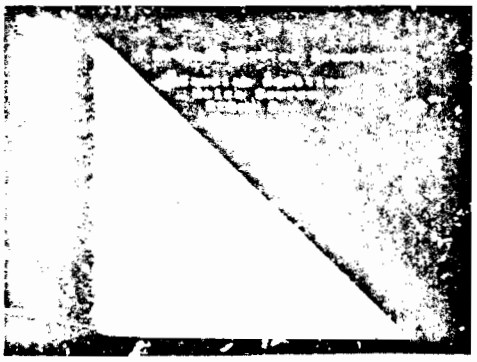


Figure 5:
Graph of Probability of
boundary vs Ed Gradient
For Gaussian Additive Noise
(stdev 8 out of 256)

Intensity is proportional to probability of a boundary (White means boundary Black means no Boundary) From left to right is successively higher gradients. From bottom to top is successively brighter windows.

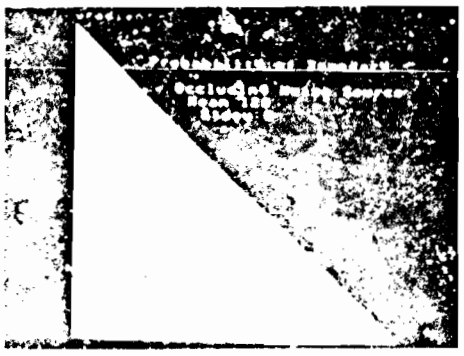


Figure 6
Graph of Probability of
Boundary vs Ed Gradient
For Occluding Noise
(mean 128 (out of 256) stdev 8)

Intensity is proportional to probability of a boundary (White means boundary Black means no Boundary) From left to right is successively higher gradients. From bottom to top is successively brighter windows.

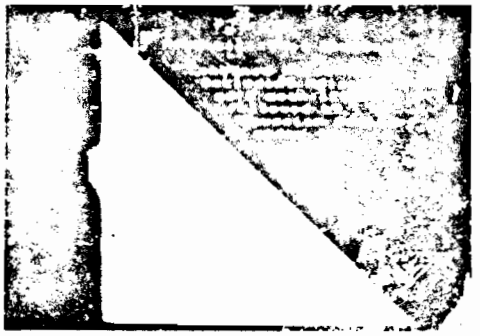


Figure
Graph of Probability of
boundary vs Ed Gradient
For Occluding Noise
(mean 128 out of 256) stdev 8)
Followed by Gaussian Additive Noise
(stdev 4)

Intensity is proportional to probability of a boundary (White means boundary Black means no Boundary) From left to right is successively higher gradients. From bottom to top is successively brighter windows.

In these figures I graph the probability of a boundary passing through a two pixel window, against the gradient and the minimum gray-level of the pixels in the window the gradient is calculated from. Figure 5 graphs the probability of a boundary assuming gaussian additive noise. Note that the probability of a boundary depends only on the gradient and not on minimum intensity in the window. Figure 6 shows the same graph assuming just occluding noise. Here, for nonzero gradient the only variation in probability is because of the differing minimum intensities in images with windows containing pixels likely to be noise having a lower probability of containing a boundary. Figure 7 shows the same graph with a more realistic assumption of occluding noise followed by gaussian additive noise.

12. End of Summary

I have just summarized how a low level system for image interpretation can be made from human knowledge in a flexible way that does not ignore significant portions of either the *a priori* known data or the observed image. I have summarized the result of testing some of these techniques using simple domains. Research on taking into account more varied sources of information and finding realistic priors is underway. Also the techniques for building domains and analyzing operators are to be applied to more sophisticated models and operators.

References

1. J. Aloimonos and P. Chou, Detection of Surface Orientation and Motion from Texture: 1. The Case of Planes, 161, Computer Science Department, University of Rochester, January 1985.
2. H. C. Andrews and B. R. Hunt, *Digital Image Restoration*, PRENTICE-HALL, INC., Englewood Cliffs, New Jersey 07632, 1977.
3. H. C. Andrews and B. R. Hunt, *Digital image Restoration*, PRENTICE-HALL, INC., Englewood Cliffs, New Jersey 07632, 1977.
4. D. H. Ballard and C. M. Brown, in *Computer Vision*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1982, 102 105.
5. D. H. Ballard and O. A. Kimball, Rigid Body Motion from Depth and Optical Flow, 70, Department of COmputer Science, University of Rochester, November 1981 VISION.
6. R. Chellappa, Fitting Markov Random Field Models to Images, 994, University of Maryland, COmputer Vision Laboratory, Computer Science Center, January 1981.
7. R. Chellappa, Digital Image Restoration using Conditional Markov Models, 1027, University of Maryland, COmputer Vision Laboratory, Computer Science Center, March 1981.
8. H. Derin, H. Elliott, R. Cristi and D. Geman, Bayses Smoothing Algorithms for Segmentation of Binary Images Modeled by Markov Random Fields, *PAMI 6*, 6 (Novermber 1984), 707-720, IEEE.
9. D. J. Fleet, The Early Processing of Spatio - Temporal Visual Information, 84-7, University of Toronto, Research in Biological and COmputational Vision, September 1984.
10. S. Geman and D. Geman, Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images, *PAMI 6*, 6 (Novermber 1984), 721-741, IEEE.
11. A. R. Hanson and E. M. Riseman, VISIONS: A Computer System for Interpretting Scenes, in *Computer Vision Systems*, A. R. Hanson and E. M. Riseman (ed.), Academic Press, London, 1978, 303-334.
12. B. K. P. Horn, *Shape from Shading: A Method for FInding the SHape of a Smooth Opaque Object from One View*, Massachusetss Institute of Technology Department of ELectrical Engineering, August 1970.
13. K. Ikeuchi, Shape form Regular Patterns (an Example of Constraint Propagation in Vision), 567, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, March 1980.
14. T. K. J. R. Kender, Mapping Image Properties into Shading Constraints: Skewed Symetry, Affine - Transformable Patterns, and the Shape from Texture Paradigm, 133, Carnegie Mellon University Computer Science Department, July 1980.

15. G. Reynolds, N. Irwin, A. R. Hanson and E. M. Riseman, Hierarchical Knowledge-Directed Object Extraction Using a Combined Region and Line Representation, *Proceedings: Image Understanding Workshop*, , October 1984, 165-168.

16. D. Terzopoulos, Multilevel Computational Processes for Visual Surface Reconstruction, *Computer Vision Graphics, and Image Processing 24*, 1 (October 1983), 52-96, Academic Press.

GC970145-32

LOCATING CULTURAL REGIONS IN AERIAL IMAGERY USING GEOMETRIC CUES

Pascal Fua and Andrew J. Hanson

Artificial Intelligence Center
SRI International, Menlo Park, California

ABSTRACT

To locate cultural regions in aerial imagery, we merge pixel-level techniques with geometric reasoning and generic (as opposed to specific or template-like) object descriptions. We utilize discrepancies between the generic models and the image data to refine an initial low-level segmentation and produce a more accurate delineation of cultural regions.

1 Introduction

Detecting and labeling scene objects is one of the more demanding tasks in automated image analysis. In the typical case of a high-altitude aerial image, there are no existing segmentation techniques that can reliably produce regions that have a one-to-one correspondence with objects of interest. Most segmentation procedures produce a wide mixture of undersegmented objects, where the object is merged with other data, and oversegmented objects, where the object is broken up into a "jigsaw puzzle" of indistinct parts. Furthermore, such segmentations are normally unstable with respect to minor changes in the program parameters, digitization methods, viewpoint, scene lighting, and film-processing methods.

We therefore propose to explore the application of knowledge-based methods to the problem of correcting an initial segmentation so it coincides with recognizable objects. Other related efforts include those of Ohta et al [1979], Nagao et al [1980], Reynolds et al [1984], Nazif and Levine [1984], McKeown et al [1984], and Hwang et al [1985]. Our work relies upon contextual geometric reasoning and generic, template-free models of the features to be extracted from the image. We overcome some of the limitations of previous approaches by providing powerful facilities for utilizing generic shapes and spatial context to resolve undersegmented objects.

The work reported here was supported by the Defense Advanced Research Projects Agency under Contract MDA903-83-C-0027 and by the U.S. Army Engineer Topographic Laboratories under Contract DACA72-85-C-0008.

For the purposes of our current work, we have imposed the following constraints:

- **Object type:** We restrict ourselves to the identification of cultural structures in aerial imagery, thereby providing the opportunity to use such observations as the presence of straight lines to focus attention on regions likely to be components of a target object [see, e.g., Shirai, 1978].
- **Image data:** We assume that we are given a digitized aerial image that is essentially a straight-down view, along with lighting and camera-model parameters. Typical images used in our experiments have scales of 1 to 2 feet per pixel on the ground.
- **Initial segmentation:** We assume we are provided with a syntactic partition of the image computed by an Ohlander-style segmenter [Ohlander et al, 1978; see also Laws, 1982, 1984].
- **Knowledge characteristics:** We assume that no precise templates of the target cultural objects are available, and thus we are required to deal with complex objects having only general, semantic descriptors.

Our results to date may be summarized as follows:

- **Undersegmented Regions Are Correctly Refined.** The identification of cultural portions of a region on the basis of groups of parallel and perpendicular lines leads to a very reliable splitting of undersegmented regions when combined with other contextual knowledge.
- **Templates Are Eliminated.** Many traditional systems for discovering buildings use relatively rigid rectangular templates, possibly with an allowable range of constraints on dimensions [e.g., Binford, 1982; Hwang et al, 1985]. Instead, we employ *generic* knowledge of the object geometry. By generalizing the concept of a "side" to include a large class of rectilinear zig-zag shapes and searching for rectangular geometric relationships among these compos-

ite shapes, we can accept and identify very complex polygonal structures with rectilinear components. No assumptions whatsoever are made about *specific* shapes, and thus we avoid the restrictions of the template approach while gaining substantial power.

- **Semantic Knowledge Supports Correction and Labeling of the Initial Segmentation.** We have linked domain knowledge with image-level operations in several ways to improve overall system behavior. We utilize knowledge of how the segmenter is likely to misplace region boundaries relative to desirable edges to recover such edges in the resegmentation, as well as to reject improbable geometries. Predicting the way shadows may be separated or incorrectly merged in the original segmentation leads to the correct parsing of shadow evidence required for identification of raised structures.

In the succeeding sections, we first describe our general approach to the design of an object-recognition system, and then present some initial results. We conclude with our plans for future refinement of the system

2 Approach to the Object Recognition Problem

Several observations and theoretical concepts form the basis for our approach to the object recognition problem.

Recursive segmentation guarantees strong derivatives. An Ohlander-style segmentation of an image is recursive. A set of pixels in a given value range is selected on the basis of the shape of a frequency-of-occurrence histogram; these pixels are then labeled as belonging to one of several regions on the basis of spatial contiguity. The histogram of a region derived in this way will often have a shape entirely different from the parent histogram. The procedure is applied recursively until regions with no significant histogram structure are obtained.

Neighboring regions thus will often belong to *noncontiguous* value ranges of the histogram; the deeper the level of recursion, the more likely it is to find regions widely separated from their neighbors with respect to the range of pixel values in their histograms. *Region boundaries tend to lie on discontinuities in the pixel values and, therefore, strong derivatives occur between regions*

In Figure 1, we verify these observations for a grey-scale image by showing the qualitative correspondence between segmentation region boundaries and the pixels in the image with high Sobel derivative strengths.

Sobel directions align with region boundaries. Edge direction can be determined in two ways. One is to fit a line to a set of points in an edge sequence, and the other is to compute the Sobel direction at a point. Because of the high correlation between Sobel derivatives and region boundaries shown in Figure 1, the latter will be quite reliable (see also Burns et al, 1984, for another approach).

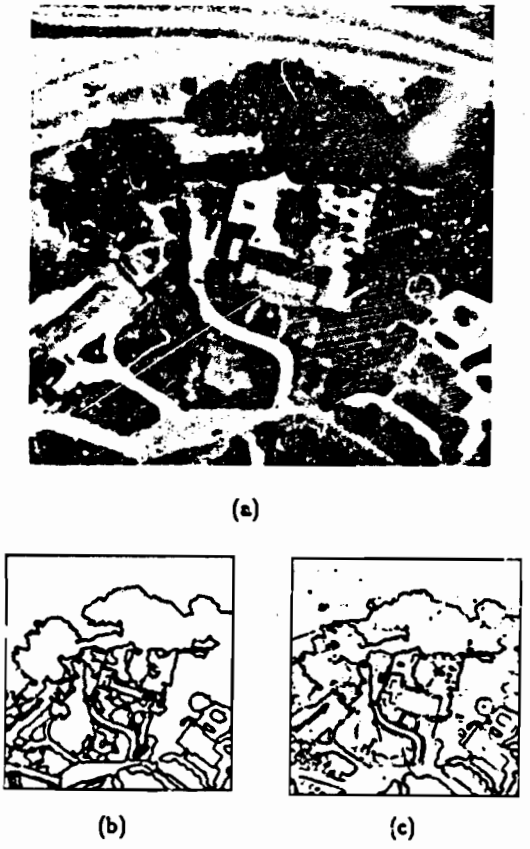


Figure 1. (a) An example of an aerial image containing houses. (b) The boundaries of the regions resulting from a segmentation of the image. (c) A binary image showing those pixels with strong magnitudes of the Sobel derivative.

In Figure 2, we show a typical region boundary obtained from the SRI SLICE segmenter [Laws, 1984], together with the long, straight lines obtained by an algorithm that looks only for consistency in the Sobel directions of a contiguous set of boundary points. The sets of points with compatible Sobel directions and the apparent linear boundary pieces are in good agreement.

Lines are classified by geometric direction. Semantically significant clusters of lines are often collinear, but *laterally displaced*. The direction that we assign to a cluster of two or more collinear or parallel lines is a

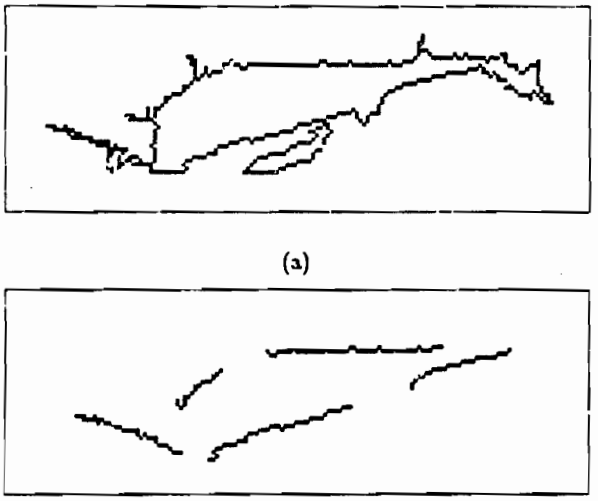


Figure 2: (a) Typical region boundary taken from the bottom center of Figure 1. (b) Long, straight lines in the region boundary derived only from requiring consistency of the Sobel directions in sets of contiguous points.

weighted average of the directions of each individual line, rather than the direction produced by fitting a line to the complete collection of points. This distinction is illustrated in Figure 3.

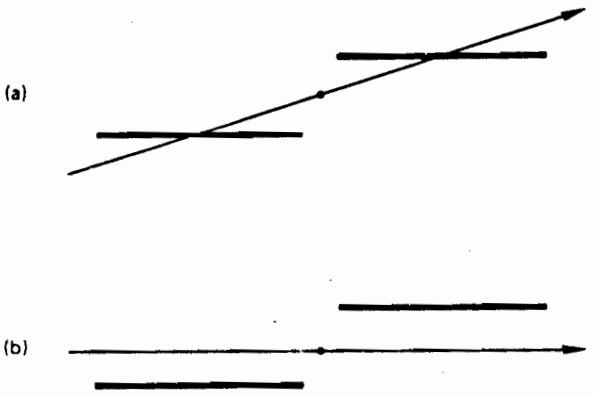


Figure 3: (a) The result of fitting a line to all the points in a pair of parallel, offset lines. The resulting direction is *incorrect* for the purposes of this work. (b) The composite direction of two lines computed from a weighted average of the direction of each line.

Shadows may be separated efficiently. Shadows form high-contrast regions with predictable geometric shape characteristics [see, e.g., Shafer, 1985; Medioni, 1983]. Our line-extraction methods are especially appropriate for extracting shadows that may have several broken segments aligned with the sun azimuthal angle.

Backtracking mechanisms are supported. Backtracking is accomplished in the current system using a library of reversible, rule-like procedures. An example of such a backtracking operation is shown in Figure 4; a composite line can be broken when a rule gives preference to the construction of a more complex structure, such as a U-shape.

We have previously expressed portions of our system in the framework of MRS [Geneserith et al, 1983] in an attempt to utilize the backtracking facilities provided in such a reasoning environment; in the current implementation, we have chosen for practical reasons to revert to procedural rule representation. Perhaps when a more complete understanding of this problem domain is achieved, we shall translate some of our procedurally represented rules into a more succinct declarative representation.

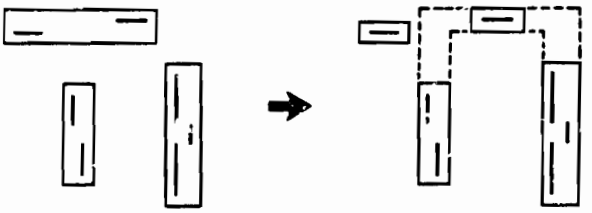


Figure 4: Backtracking by breaking a composite line to form a U-shaped structure. The U-shape is preferred because it provides strong evidence for a cultural object.

Geometric structure localizes semantically significant subregions. The current system relies upon general relationships such as perpendicularity and parallelness of composite line structures to single out portions of an arbitrarily shaped region that have suggestive polygonal substructures. This information is then used to correct the original segmentation

We extract and use relationships such as *in front of, behind, between, beside, enclosed by, enclosing, at a certain angle from,* and *at a certain distance from* in both geometric and contextual reasoning processes. This vocabulary provides a basis for semantic reasoning, e.g., "look for dark areas in the direction of the solar azimuthal angle relative to a region boundary in order to confirm the hypothesis of a building wall."

Once interesting region portions are selected, a pixel-based line-linking procedure can be invoked to connect related lines, complete corners, and close open-ended Parallels or U's. When the resulting links are satisfactory, the undesirable portions of the region are amputated, leaving clean cultural structures as the residue. Figure 5 illustrates linking processes that would be carried out when significant linear structures are present in an undersegmented region.

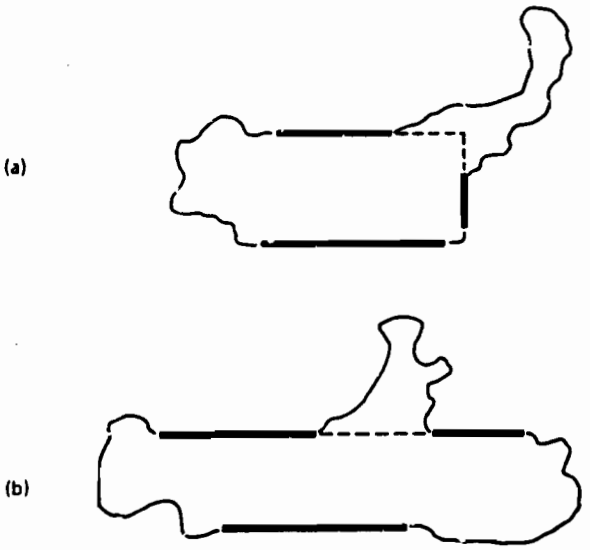


Figure 5: (a) Resegmenting a region with a good U by completing a corner. (b) Resegmenting a region with a good Parallel by linking the elements of a composite line.

3 Examples and Results

The current implementation of the system consists of two main sequences of operations:

- Discovering the geometric features and relationships within each single region.
- Resegmenting some regions based upon geometric relationships within a region or among distinct regions, and grouping interesting regions based on context knowledge.

Resegmentation is currently carried out using the F^* algorithm of Fischler et al [1981]. We compute the required cost array by using the Sobel edge strength combined with geometric constraints on the directions in which edge completion is predicted to take place. As a result, when the Sobel strength near a boundary segment follows a desirable path different from the boundary, F^* will pick up that path.

The final result of the computation is a resegmentation of the image with explicitly identified cultural-region clusters. Below, we present three examples illustrating the general features of the approach.

3.1 Example 1: An easy region.

In the lower right-hand corner of the aerial image in Figure 1a there is a house whose outline corresponds exactly to one of the regions produced by the segmentation. The good lines found in the region boundary are shown in Figure 6. This house is characterized by the two sets of parallel lines that close to form a Box; an appropriately located shadow is also present.

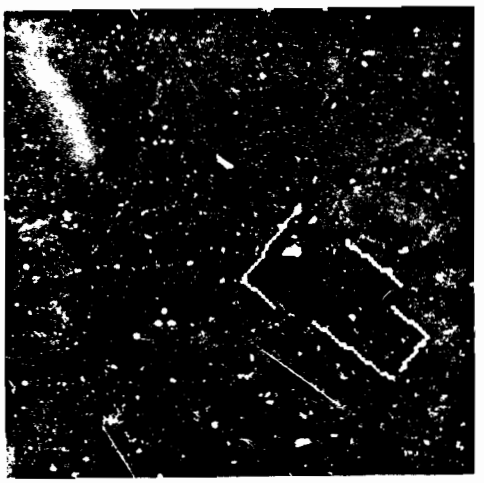


Figure 6: The long, straight lines belonging to a distinct house region. These lines form a Box structure, indicating very strong evidence for a cultural object and distinguishing the region from its surroundings.

Even when the segmentation of an image is effectively perfect, locating the cultural correspondences can be non-trivial. Our method immediately focusses on this structure without *a priori* knowledge of its shape and singles it out because of its exceptional geometric structures.

In this case, no resegmentation is performed because there is no significant difference between the paths found by linking the lines and the region boundary itself. The result is a single, identifiable house-region, as shown in Figure 7.

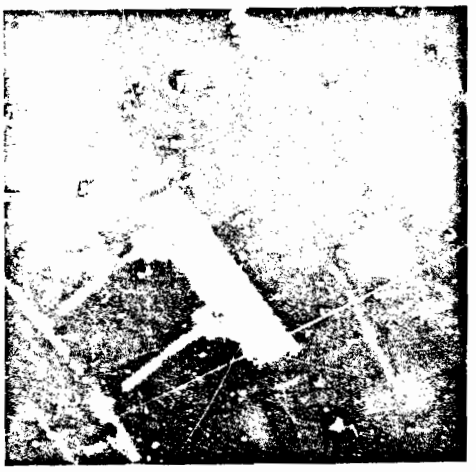
Figure 7: The single identified house region boundary overlaid on the image. No resegmentation was necessary in this ideal case.

3.2 Example 2: Repartitioning a complex region

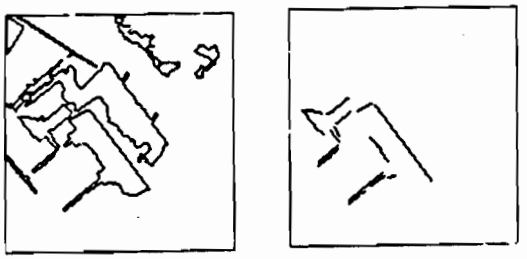
The next example, with the image, region boundaries, and elementary line segments shown in Figure 8, contains a heavily shadowed, approximately L-shaped, composite building. The segmentation confuses complex porches with roof tops, inappropriately combines sidewalks with the roof, and merges a significant shaded roof portion with background vegetation. The sunlit portions of the composite roof are contained in a single region; the two main lobes of this region are joined by a narrow neck. We observe that, given only the good edges of this roof region as shown in Figure 8c, the roof structure is confusing to parse even for a human.

We first search for basic geometric relationships within the roof-containing region. Two distinct U's are found that support the identification of a cultural object, one in each lobe of the region. Both of these U's require the breaking of a composite T, a type of backtracking, for their construction.

Next, the system attempts to link composite lines and to close the open ends of the U's to form boxes using the line-linking algorithm. This procedure amputates the porch and sidewalk appendages and leaves two Boxes, outlined in Figure 9, that provide a clear semantic context. Applying knowledge of shadows here generates the hypothesis that both boxes are associated with the same large shadow region, so we label the group as a composite 3-dimensional structure.



(a)



(b) (c)

Figure 8: (a) Another image containing a complex house structure. (b) Region boundaries. The upper L-shaped structure is a shadow; the lower L-shaped structure arises from two juxtaposed pieces of sunlit roof joined into a single region by a narrow neck. (c) Long, straight lines in the boundary of the sunlit roof region.

We have thus succeeded in taking a single, confusing region and using its geometric structure to break it up into manageable parts. We note that the area enclosed between the pair of Box structures and the shadow is a heavily shaded, peaked-roof portion whose region features are so poor that it could not have been recognized by our basic methods; the labeled enclosing regions now provide the required semantic context to support this identification.

3.3 Example 3: Multiple region clustering

In Figure 10, we show another portion of the image of Figure 1a and its segmentation. This image is typical of cultural scenes that are difficult to parse using pattern-

Figure 9: Final results of the splitting. The initial segmentation is split several times to give two subregions with good Box structures whose boundaries are outlined in the figure. The large shadow region is recognized as common to both subregions. The area between the shadow and the Box structures is now identifiable by its semantic context as a heavily shaded roof portion.

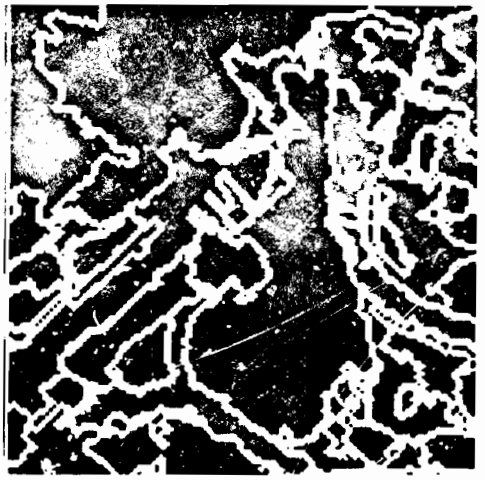


Figure 10: Left portion of the image of Figure 1a with segmentation boundaries from Figure 1b overlaid.

matching techniques because the terrain and roads are highly irregular and the houses have very complex shapes. Figures 11 and 12 show typical regions resulting from the segmentation of a house-containing area, along with illustrations of the process by which geometric structures are discovered. The first region contains an excellent U, while the second has a Parallel.

When we repeat the analysis for each region in Figure 10, we find only these two regions that have suggestive structures and appear to be geometrically related. Since an appropriate shadow region is present, we deduce that these regions probably belong to a single cultural cluster.

The geometric relations among lines in the boundaries of these regions are now used to predict the locations of the resegmentation boundaries to be constructed by linking. The results of the linking and resegmentation operations, depicted in Figure 13, show clearly the successful extraction of this complex building. We note that three different types of repartitioning were carried out to achieve this: (1) linking a corner formed by two lines belonging to a single region, thereby splitting off an irrelevant appendage; (2) linking a corner whose lines belong to two separate regions, thereby splitting yet a third region lying between them (this completes a U whose sides are the parallel lines in Figure 12); and (3) closing off the bottom of the U's formed by each of the two major roof segments.

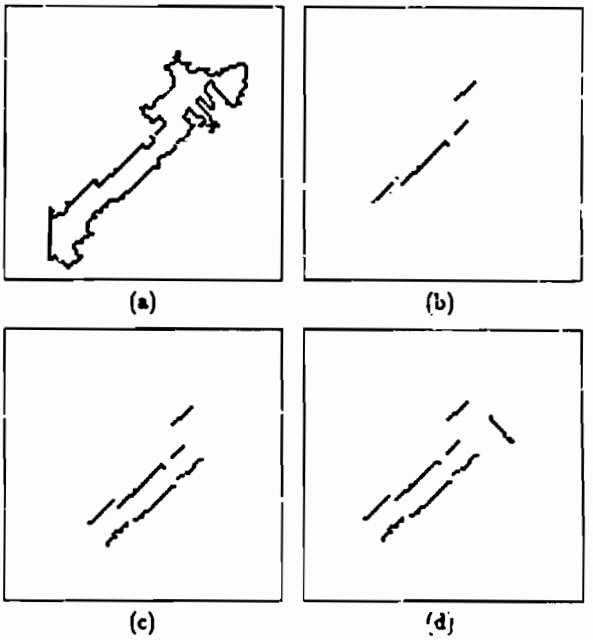
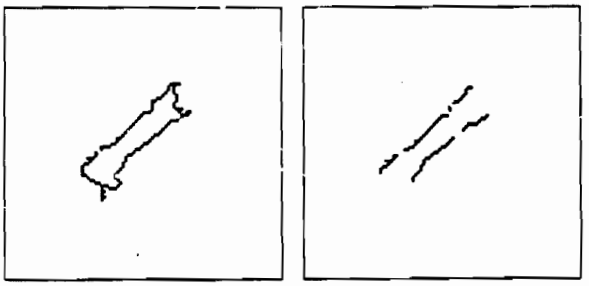


Figure 11: An illustration of the procedure by which geometrical relations are constructed within a single region. (a) Boundary of a typical region including portions of a house. (b) An example of a composite line with many elementary components extracted from the boundary. (c) A pair of parallel lines formed within the boundary by two composite lines. (d) The U structure constructed by finding a line in the boundary that closes off one end of the parallels. In this example, all good line segments belong to the U.



(a) (b)

Figure 12: (a) Border of a second region belonging to the same house. (b) These parallel lines are the best structure that can be built. The Sobel directions of the short left edge are not sufficiently consistent to allow us to accept it as a closing line for a U structure.

The roof segments are labeled as belonging to a 3-dimensional, raised structure with a peaked roof, since they correspond to a "sunny side" and a "shady side" of the roof, with a narrow shadow adjacent to the "shady side."

4 Directions for Future Work

We plan to add the following enhancements to the current system during the next stage of development:

- Generate interactive explanations of various actions to facilitate user understanding and debugging of domain rules; support user input of domain knowledge and corrections of the labeling.
- Merge "jigsaw puzzles" of objects that have been badly oversegmented.
- Extend the domains of expertise to include "explainable anomalies," of which the current shadow analysis is one example.
- Support additional classes of target objects.
- Incorporate additional geometric information such as perspective distortion of target shapes present in oblique views and nonplanarity of the underlying land.
- Support exploitation of multiple images covering the same scene.

The investigation described here explores a number of promising theoretical directions for knowledge-based partitioning and object identification, and produces satisfying experimental results for particular classes of images. Our next task will be to extend these ideas while incorporating support for explanatory interactions with the user.

Figure 13: The results of computing linking lines and cutting regions accordingly. A third region comes into play when the linker completes the right-hand corner. The resulting three regions contain the area that one would visually associate with a house.

References

T.O. Binford, "Survey of Model-Based Image Analysis Systems," The International Journal of Robotics Research, 1, No. 1, pp. 18–64 (Spring, 1982).

J.B. Burns, A.R. Hanson, and E.M. Riseman, "Extracting Straight Lines," Proceedings of the Image Understanding Workshop, pp. 165–168 (1984).

M.A. Fischler, J.M. Tenenbaum, and H.C. Wolf, "Detection of Roads and Linear Structures in Low-Resolution Aerial Imagery Using a Multisource Knowledge Integration Technique," Computer Graphics and Image Processing 15, pp. 201–223 (1981).

M. Genesereth, R. Greiner, and D.E. Smith, "MRS — A Meta-Level Reasoning System," Stanford University Heuristic Programming Project Report No. HPP-83-27 (1983).

V. Hwang, L. Davis, and T. Matsuyama, "Hypothesis Integration in Image Understanding Systems," University of Maryland Report CAR-TR-130 (1985).

K.I. Laws, *The PHOENIX Image Segmentation System: Description and Evaluation*, Technical Note 289, Artificial Intelligence Center, SRI International, Menlo Park, California (September 1982).

K.I. Laws, *Goal-Directed Texture Segmentation*, Technical Note 334, Artificial Intelligence Center, SRI International, Menlo Park, California (September 1984).

D. McKeown, W.A. Harvey, and J. McDermott, "Rule Based Interpretation of Aerial Imagery," Proceedings of IEEE Workshop on Principles of Knowledge-Based Systems, pp. 145–157 (1984). See also IEEE Trans. PAMI, *in press.*

G.G. Medioni, "Obtaining 3D Information from Shadows in Aerial Images," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 73–76 (1983).

M. Nagao and T. Matsuyama, *A Structural Analysis of Complex Aerial Photographs*, Plenum Press (New York, 1980).

A.M. Nazif and M.D. Levine, "Low Level Image Segmentation: An Expert System," IEEE Trans. PAMI 6, pp. 555–577 (1984).

R. Ohlander, K. Price, and D.R. Reddy, "Picture Segmentation Using a Recursive Region Splitting Method," Computer Graphics and Image Processing 8, pp. 313–333 (1978).

Y. Ohta, T. Kanade, and T. Sakai, "A Production System for Region Analysis," Proc. 6th Inter. Joint Conf. on Artif. Intell., pp. 684–686 (1979).

G. Reynolds, N. Irwin, A.R. Hanson, and E.M. Riseman, "Hierarchical Knowledge-Directed Object Extraction Using a Combined Region and Line Representation," Proceedings of the Image Understanding Workshop, pp. 195–204 (1984).

S.A. Shafer, *Shadows and Silhouettes in Computer Vision*, Kluwer Press (1985).

Y. Shirai, "Recognition of Man-Made Objects Using Edge Cues," in *Computer Vision Systems*, Ed. A.R. Hanson and E.M. Riseman, Academic Press (New York, 1978).

GC970145-33

THE INFORMATION FUSION PROBLEM AND RULE-BASED HYPOTHESES APPLIED TO COMPLEX AGGREGATIONS OF IMAGE EVENTS

Robert Belknap, Edward Riseman, and Allen Hanson

Computer and Information Science Department
University of Massachusetts
Amherst, Massachusetts 01003

ABSTRACT

A rule-based system for combining information from multiple sources of sensory data is described. Relational rules, integrating data from the output of multiple low level processes, are responsible for creating complex aggregations of the data in order to obtain object hypotheses with an associated confidence. Relational rules are defined between primitive elements of the data abstractions so that sets of elements across representations can be selected and grouped on the basis of relational scalar measures. The system is demonstrated using region and line data and a set of relational measures defined over the two pixel-based representations. The techniques presented are extensions of an earlier rule-based system operating on single types of data abstractions and are easily extended to include motion, stereo, and range data.

I. INTRODUCTION

The problem of integrating information from multiple low-level representations is just beginning to be investigated in computer vision. In the past this had not been a problem, since most systems dealt with only one type of low-level or segmentation information (usually lines or regions), and this information was extracted from only one type of sensory input data (usually intensity or RGB). The field of computer vision has matured and now there are many applications where multiple low-level processes provide partially non-redundant and useful information.

The problem of integrating multiple representations of data can occur even when there is only one source of sensory data. Different algorithms applied to the same data will extract different feature events. For example, algorithms for extracting straight lines often will provide information that complements a region segmentation. Of course, the utility of multiple representations is dependent upon the particular set of algorithms employed, the task domain, and the goals.

This work has been supported by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. N00014-82-K-0464, and by the Air Force Office of Scientific Research under contract F49620-83-C-0099.

In addition to different types of information extracted from the same data, the use of multiple sources of data is becoming more common. Depth maps are being obtained directly from laser range data and indirectly from motion and stereo algorithms applied to pairs or sequences of images. These depth arrays are sources of additional intermediate representations, such as surfaces or three-dimensional lines at discontinuities of surface orientation. Certainly different information will be obtained from intensity arrays and depth arrays, and therefore representations derived from depth maps provide information which is important to integrate. It has become obvious that each low-level process usually extracts only partial information with a great deal of redundancy in these outputs; thus, maximum reliability can only be achieved through processes that can integrate information in a flexible manner.

This paper describes one aspect of this problem, that of integrating information from multiple low-level processes applied to multiple sources of sensory data to obtain a single object hypothesis with an associated confidence. The approach described in this paper is illustrated via regions and lines extracted from the same RGB data. The techniques presented, however, could easily be extended to include motion, stereo and range data which yield an intermediate representation of surfaces. The approach also will naturally apply to motion attributes of lines, regions, surfaces, and volumes which have 2D or 3D motion attributes.

II. BACKGROUND

II.1. THE INTERMEDIATE SYMBOLIC REPRESENTATION

It is generally accepted that a computer vision system must perform a variety of transformations of the data during the interpretation process. One of the key abstractions is the transformation of pixels, or more generally arrays of sensory data, into image events which can be named and referred to by their properties. We refer to the resultant representation as "the intermediate symbolic representation". This representation serves as the communication interface between low and high level processes in the VISIONS system [HAN78a, RIS84].

At a coarse conceptual level, the VISIONS system is organized into three levels of processing: low, interme-

diate, and high. Currently, the low-level, or segmentation, processes output a symbolic representation of the data in the form of regions and lines. Attributes, such as color, texture, location, size, shape, and orientation, are then calculated for each region or line. All of this information is then organised into an intermediate data structure [SOU85] which allows flexible access to attributes of, and relations between, the intermediate symbolic entities. The intermediate-level processes organize the low-level output and produce initial hypotheses (for the high-level processes which interpret the scene) through the use of domain knowledge and provide control information to modulate the lower level processes.

In general, the only requirement for placing another type of low-level entity in the intermediate representation is that each primitive element of that data type must have a symbolic name (e.g., region-240, surface-38, corner-46) and a non-empty set of attribute-value pairs. It is the values of the attributes that provide the basis for initial interpretation processes. At present the intermediate database consists of regions and lines.

In the next two subsections we will very briefly outline the low-level segmentation algorithms, and then the current version of the rule-based system for generating hypotheses from single types of intermediate elements. Succeeding sections extend this system to operate over multiple types of intermediate elements.

II.2. SEGMENTATION ALGORITHMS

We have developed our region and edge representation based on two low-level algorithms: the Nagin-Kohler region segmentation algorithm [NAG79, NAG82, KOH83, KOH[illegible]] and the Burns linear feature extraction algorithm [BUR[illegible]]. The Nagin-Kohler algorithm invcives the detection of clusters in a feature histogram, associating labels with the clusters, mapping the labels onto the image pixels, and then forming regions of connected pixels with the same label. In general, the process of global histogram labeling causes many errors to occur, hence an additional key aspect of this algorithm is the localisation of the histogramming and peak selection to subimages. Regions are formed from the local histogram labels and then the results in subimages are integrated via region merging across the artificial subimage boundaries.

The Burns algorithm is directed toward the extraction of linear features in Intensity images, including low-contrast linear features. It provides a low-level representation of intensity variations by segmenting the intensity array into connected subsets of pixels which have similar gradient orientation; these pixels act as "edge-support regions" of a linear feature, and various characteristics of the associated line or edge can be extracted from them. Thus, both regions and lines have a common pixel based representation which is a significant advantage for information integration.

II.3. OBJECT HYPOTHESES VIA RULES ON ATTRIBUTE RANGES

A simple type of knowledge source for generating hypotheses of object class labels for particular regions has been under development in the VISIONS environment [WIL81, WEY83, RIS84]. They take the form of simple rules defined in terms of ranges over a scalar feature, and complex rules defined as combinations of the output of a set of simple rules. The scores of these rules serve as a focus of attention mechanism for other, more complex knowledge-based processes. The rules can also be viewed as sets of partially redundant features each of which defines an area of feature space which represents a "vote" for an object on the basis of this single feature value. The region attributes include color, texture, shape, size, image location, and relative location to other objects. More recently, the approach has been extended to lines, with features including length, orientation, contrast, width, etc. In many cases, it is possible to define rules which provide evidence for and against the semantically relevant concepts representing the domain knowledge. While no single rule is totally reliable, the combined evidence from many such rules should imply the correct interpretation. One additional rule type is the intersection rule which allows line information to affect the score of a region or vice-versa.

The simple rules consist of a feature and a piecewise-linear mapping function specifying veto ranges and a central positive voting range, with zero voting ranges around it (see Figure 1). The mapping function is defined by six parameters specifying values on the range of the feature and correspond to "veto", zero score, maximum score, maximum score, zero score, and "veto" in that order. As Figure 1 shows, this format is meant to approximate a Gaussian shaped function.

An interactive environment for constructing these rules and displaying their effects has been created. A user can write a rule and display the scores of either regions or lines in an intensity coded format. The system then allows the user to edit the rule or to incorporate it into a complex rule.

Complex rules are an hierarchical collection of other rules. These rules could be simple or complex rules. We have typically structured the top-level complex rule for an object as a set of five other complex rules which represent color, texture, size, shape, and location rules. The scores of these separate component rules are usually combined with a weighted average function, although any numeric combination of the scores can be used. A typical top-level object hypothesis rule is shown in Figure 2. Complex rules can be written and edited in the same way as simple rules [BEL85].

II.4. BINARY RELATIONS AND THEIR USAGE

The rules described in the previous section are unary -- they accept a region as input and return a confidence for the hypothesised object. The highest ranked of these hypotheses form a partial (and perhaps errorful) interpretation of the original image. Rules may be defined over pairs of regions; rules of this type define a binary relation between the regions and the confidence values returned by the rule specifies the degree to which the relation holds. Binary rules can begin to capture some of the contextual constraints which the developing interpretation is expected

to satisfy.

In the current implementation of the rule system, the only binary rules are similarity rules [HAN85], which are used to form aggregations of regions with similar properties. In effect, the unary rule hypotheses form "islands of reliability" which are extended using binary similarity rules. Typically, these rules operate on primitives formed by a single segmentation process (e.g., regions or lines), and result in a merging of the primitives into a more complete description, depending on the confidence returned by the rules. Forming aggregations of elements from a single segmentation process has advantages when dealing with unreliable segmentation processes. Fragmented elements can be grouped to form aggregates which match an object model. The problem with this approach is that unguided grouping of elements leads to a combinatorial explosion. Section III.5 outlines a method of forming aggregates under the guidance of other segmentation information, which would significantly reduce the number of possible aggregations.

Plausible groupings (perhaps with non-empty intersections) of both regions and lines can be generated using binary relations. Regions can be grouped on similarity of multiple attributes, such as color, texture, and adjacency as well as on the degree to which the grouping satisfies some unary constraint, such as shape (does the hypothesized grouping form a rectangle?). Relations that could be used to group lines include similarity of orientation and proximity of endpoints [e.g., WES85]. Pairs or sets of parallel lines could be grouped. Pairs or sets of lines whose endpoints fall in the same local neighborhood could be grouped. Each of these groups could then be stored as an entity with its own set of attributes. Matching groups of lines to a model would be much more reliable than matching based in the attributes of a single line. Objects with a definite shape, such as rectangular, could be extracted from the groups of lines. Also, objects with characteristic lines, such as the many parallel lines found in a road, could be extracted using lines only.

Binary relations also permit the fusion of information from multiple segmentations, or more generally from multiple sources of information. We explore this issue in Section III.

II 5. RELATED RESEARCH

There have been a few efforts to integrate results from multiple low-level processes [HAN78b, KOH84, NAS83]. Usually these have involved the integration of line and region data, which are the two most common types of low-level algorithms employed.

Kohler [KOH84] proposes solutions to the information fusion problem at the segmentation level. The first method proposed is to allow each process to segment the image independently. Regions from each segmentation are then combined in one of two ways: 1) project all region boundaries onto one image and merge the resulting regions, or 2) project only those boundaries with support from all segmentations and connect the resulting boundary segments. The second method is to allow each segmentation process to influence other segmentation processes during the initial segmentation.

Nasif and Levine [NAS83] also address the problem of integrating multiple sources of information at the segmentation level. Their system uses a rule-based expert system to segment natural scenes based on a region and line representation. Production rules are used to split and merge regions and lines based on spectral attributes and binary relations between regions and lines. An additional set of rules is used to group regions and lines into focus of attention areas which are used to guide the application of segmentation processes.

Reynolds et al. [REY84] use region and line information to guide processing in a hierarchical interpretation of very large aerial images. Lines which bound regions in a coarse-level segmentation are used to locate regions with rectangular outlines. The parameters of long lines with high contrast are mapped through a Hough space to find a peak of common orientation. The lines which correspond to this peak and the regions with rectangular outlines are used to select interesting areas of the image, which are then subjected to finer levels of segmentation.

Bajcsy and Tavakoli [BAJ76] present a system for recognizing roads from aerial images. In their system a world model of roads is used to extract road segments as elongated regions. These segments are then grouped based on orientation and proximity to form road hypotheses

Mckeown, Harvey, and McDermott [MCK84] do map-guided interpretation of aerial images by applying prototype rules to regions. These rules build initial class or sub-class hypotheses for each region. Regions of a common class or sub-class are then grouped and the prototype rules are reapplied to the groups. This process continues until acceptable functional areas are defined for the image.

III. INTEGRATING REPRESENTATIONS VIA COMPLEX AGGREGATES

The fundamental problem that is being addressed in this paper is the integration of multiple low-level representations into the interpretation process. While the approach presented here offers only one type of mechanism and deals with only some of the most general levels of the information fusion problem, there are several important advantages. First, it extends an approach that has proven to be somewhat effective on very complex natural scenes [RIS84]. Secondly, it offers an entirely modular and natural approach to incorporating additional processes and representations as a vision system undergoes incremental development; existing low-level representations do not have to be modified in any way. Each low-level representation exists independently of any other representation; the integration is accomplished through the hypotheses created by rules which relate entities in the independent representations. Finally, the integration takes place at the interpretation level, thereby bridging the gap between low and high level processes.

The key integrating mechanism is the definition of re-

lations, with associated computational measures, between primitive elements in each type of representation. Each relation is expressed in the form of a rule called a *relational rule*. These are defined between "primitive" elements so that sets of elements across representations can be selected and grouped on the basis of relational scalar measures. The result, therefore, will be to extend our representation to include complex aggregations of elements which satisfy user-specified relations across the multiple representations.

Let us be somewhat more specific. Consider, for example, the interpretation of a road scene on the basis of an intermediate representation of lines, regions, and surfaces (planar surfaces) with attributes such as orientation, height off the ground plane, and distance. The formation of a "road" object hypothesis should not be based upon any single element, but rather upon an aggregation of a set of lines, regions, and surfaces that have specific relations to each other. One might set up a rule-based strategy for searching for a region which is approximately bounded by straight lines on either side, with a surface that significantly overlaps the region, whose orientation is approximately perpendicular to the gravitational normal or parallel to the ground plane. At this point we wish to caution the reader not to be misled into an oversimplified view of the problem; there are extremely difficult issues to be dealt with, such as fragmentation of lines, regions, and surfaces or lines and surfaces that are only partially consistent with a given region. However, these are problems that are implicit in the nature of the problem of integrating unreliable and inconsistent information and will be true of all approaches, not just the approach presented here.

In the sections that follow we will develop this approach as applied to the output of region and line segmentation processes that have been developed over several years and which are currently available in the VISIONS system. We are working on the extraction of surface elements from depth maps produced by motion and stereo algorithms, and when these are available we will extend our representation to allow surfaces to be aggregated with regions and lines.

III.1. REPRESENTATION OF REGIONS AND LINES

Before we discuss the development of relations between lines and regions, let us briefly discuss the representations of these two types of elements. Regions are by definition a pixel-based representation because they are connected sets of pixels. The line representation is somewhat unusual and allows a very simple foundation for the line-region relations. Lines in our algorithm are each extracted from their line support region, which is the portion of the intensity surface (i.e., set of pixels) that forms the intensity discontinuity defining the line. The important point is that lines are actually defined by a region, and therefore both segmentation processes result in a pixel-based representation. The natural duality between regions and their boundary lines can be exploited in a straight forward manner, since the line-support region associated with a linear feature should overlap the regions on either side of the region boundary. This means that a set of line support regions can be expected to be found superimposed on the boundary of an intensity region. Conversely, a set of adjacent intensity regions can be associated with a line support region of a linear feature. Because each line is defined by a line support region, lines and regions which don't intersect but are close to each other will still be included in the relational measures. Since the the width of the line support region reflects the width of the edge, the intersection of a line support set with a region is a fairly accurate indication that it is related to the region. Several examples are shown in Figure 3.

III.2. MEASURES OF THE RELATION BETWEEN LINES AND REGIONS

The simultaneous use of both region and line information permits two types of perceptual grouping processes to take place. On the one hand a region or set of regions can guide the grouping of lines, while on the other hand a line or set of lines can guide the grouping of regions. In the following discussion, we use the terms region and line to refer to either the primitive elements obtained from the corresponding segmentation process or to sets of regions and lines already grouped by some other process, such as the similarity grouping described earlier.

Using regions as the primary element by which lines are grouped leads to two types of measures: texture and shape. Simple texture measures can be derived easily. All lines within a region are filtered, histogrammed, or counted in order to produce a linear texture measure for the region. This method is useful for regions with some type of regular texture, such as short lines of similar orientation or contrast. Shape characteristics of a region can be derived from the lines which fall on or near the boundary of the region. Some primitive shape measures, such as rectangularity, can be calculated from the set of lines which bound the region. Other more complex measures of shape characteristics can also be calculated from the bounding lines.

With lines as the primary element, regions can be grouped and analysed. One possibility is to assign a *boundary type* to the line based on the attributes of the regions which fall on opposite sides of the line. Another possibility is to group all regions which fall on one side of the line as possibly belonging to the same object.

A natural basis for the relations between lines and regions is intersection. This is especially true in our system since both lines and regions are in a pixel-based representation. Intersection does, however, eliminate certain relations from the database. Segmentation elements which do not overlap will not be considered in the calculation of relations. This could lead to problems when line and/or region boundary placement is inaccurate. An alternative relational method which addresses this problem is discussed in Section III.3.

The intersecting elements of each segmentation are found by computing the intersection of each line support set with each intensity region. The lines resulting from this process fall into three categories: boundary lines, interior lines, and lines which are neither boundary nor interior and as such are not useful in calculating texture or shape measures (these are called "superfluous" lines). These three line intersection types are illustrated in Figure 3. Three measures were derived as a first pass at differentiating the

three types of line intersection. These measures are called "interior-line-percentage", "region-perimeter-percentage", and "line-length-percentage".

The first measure, "interior-line-percentage", is the ratio of line area interior to the region to total line area. Line area is calculated from the line support region obtained from the first stage of the Burns' straight line algorithm. In order to avoid confusion, the term "line support region" (originally used by Burns) will be replaced by "line support set" with the understanding that the pixels comprising this set are contiguous. Since both the line support set and the region are pixel based the interior line area is simply the number of pixels common to both divided by the total number of pixels in the line support set; an example is shown in Figure 4.

The interior-line-percentage measure discriminates interior from boundary lines but not boundary from superfluous exterior lines. An interior line will have a value of one for this measure, indicating that the line-support set is completely contained by the region. Boundary and superfluous lines which intersect will have some value between zero and one for interior-line-percentage. This indicates that the line support set crosses the region boundary and has at least one pixel in common with of the region. All lines that do not intersect the region will have a value of zero for this measure.

The second measure, "region-perimeter-percentage", is the ratio of region boundary pixels covered by the line support set to the total length of the region perimeter. Figure 5 illustrates how this measure is calculated. The third measure, "line-length-percentage", is the ratio of the length of a region boundary (i.e., the set of perimeter pixels) covered by the line support region to the total line length. That is, the number of perimeter pixels of the region which intersect the line support set, divided by the length of the line. See Figure 6 for a pictorial description of this measure. This measure was intended to indicate how much of a line actually contributes to a region boundary. A line which lies approximately on a region boundary will have a high value of line-length-percentage since its line support set will cover a length of region boundary approximately equal to its length. This measure distinguishes boundary from superfluous lines, but not superfluous from interior lines.

Region-perimeter-percentage and line-length-percentage are related in a dual representation. Region-perimeter-percentage measures the fraction of a region boundary made up of one line, while line-length-percentage measures the fraction of a line contributing to the region boundary.

The line-length-percentage measure may be unreliable for several reasons. If the region boundary is convoluted where the line intersects it the region boundary length will be too high and the measure will be inaccurate. The measure does not distinguish between lines which are parallel to the region boundary and lines which pass through the boundary at an angle. Also, lines that fall along the region boundary but do not intersect it are not included in the measures. The next section presents some alternative measures which address these problems.

III.3. ALTERNATIVE REGION-LINE RELATIONAL MEASURES

Before describing how rules are defined, we wish to briefly note a few alternative relational measures. These alternatives address the problems listed in the previous sections. These include inaccurate placement of line support sets and region boundaries, the lack of line to region boundary angle measures, and relations which are missed due to the use of intersection as the basis for the relational measures.

Line support sets are based solely on collections of pixels with common gradient direction. The algorithm to fit a line to these regions ignores small amounts of noise in the line support region and usually computes an appropriate line. The line to region relational measures presented are based solely on the line support set, rather than the extracted line, and may produce inaccurate results. One solution to this problem is to use something other than the line support set to compute the relational measures.

One approach is to use only those pixels through which a line passes to compute the same relational measures presented above. This would increase the accuracy of interior/perimeter discrimination but would lead to problems when a line lies parallel to the region boundary but does not overlap it. In this case the region-perimeter-percentage and line-length-percentage measures would be useless.

A variation of this approach is to define an artificial line support region which uniformly surrounds a line. The width of the artificial line support region could be varied according to the sharpness of the edge. This would retain the good properties of line support regions (fuzzy boundary intersection and edge width) while eliminating the problems of noisy line support sets.

A third alternative is to use the idea of chamfering [BAR78] to form distances between region boundaries and lines. A wave of spreading activation on the array can be implemented to measure distance by starting all cells (this algorithm is naturally thought of as implemented on a cellular array machine) on the line (or region boundary) with a value of 0 and propagating the field to connected neighbors while incrementing their count by 1. In t steps of propagation, cells that are a distance t away will be reached with a count of t. The cells on the receiving region boundary (or line) will have their distance determined by the earliest marker (i.e., lowest value) or by some function of the values resident in the cells corresponding to the boundary of line. Thus, distance is obtained naturally, and various other geometric relations, such as intersection, parallelness etc. can can also be measured. This approach captures many of the measures we have just presented but does not demand the actual intersection of the two types of elements in order to extract useful information.

III.4. RELATIONAL RULES

Relational rules are the method used to integrate information from multiple representations to form complex aggregations. The relational measures presented in the preceding sections form one component of the relational

rules. The other component needed is a structured method of accessing information from each representation and the relations between elements of the representations. In our system the relational rules take the form of *intersection rules*, rules which allow access to all elements which intersect a specific element.

Intersection rules are used to combine region and line information in a uniform, flexible way. The rule type is designed to allow any arbitrary calculation based on the lines intersecting a region. For this reason the structure of the rule is somewhat complex. An intersection rule is made up of three components:

1) a *filtering rule* for selecting lines which intersect a region based on relational measures;

2) a *ranking rule* which ranks the lines which intersect a region based on line attributes; and

3) a *combination function* which calculates the final score of the region-line aggregation based on the scores from the *filtering rule* and the *ranking rule*.

The *filtering rule* is a complex line rule composed of a simple rule for each relational measure. Using the existing three relational measures a filtering rule would be composed one simple rule for each of the measures: interior-line-percentage, line-length-percentage and region-perimeter-percentage. Any of these simple rules may be omitted in which case no filtering is done on that relational measure. In this sense the filter is similar to the simple rules applied to a region or line attribute, except that it is applied to one or more of the relational measures. In most cases the filtering rule is used to veto unwanted lines so that they are not considered by the combination function.

The *ranking rule* can be a simple or complex line rule. The rule should rank each line based on how well it supports the measure being computed for the region. For example, if a texture measure is being computed those lines which represent texture (i.e., short, high-contrast) should receive a high score from the rule. A rule to determine the linearity of the boundary of a region should assign a high score to long lines.

Although the ranking rule and filter rule could easily be combined into one complex rule they have been kept separate for reasons of clarity and efficiency. By defining a separate filtering rule it is clear which lines will be included in the complex aggregation. The ranking rule represents the attributes of the lines, not the attributes of relations between lines and regions. Also, since a line may intersect more than one region the filtering rule may have to be applied multiple times to the same line. The ranking rule can be applied once for each line in the database.

The final component of intersection rules, the *combination function*, can be any arbitrary function which produces a numeric value. The combination function is supplied three inputs, the scores from the filtering rule, the scores from the ranking rule, and the intersection measures for each line. For maximum flexibility the combination function also has access to the complete intermediate representation.

These intersection rules can be used in some very diverse ways. One example is to use a filtering rule on interior-line-percentage to select only those lines which are interior to a region. The ranking rule could then be defined to select short, high-contrast lines. The score of the ranking rule could then be averaged to form a complex texture measure. Alternatively, a density measure could be calculated by counting the occurrences of lines which receive a high score from the ranking rule and then normalising by the size of the region.

As an additional example, the measure line-length-percentage could be used to select lines which lie mostly on the boundary of the region. The ranking rule could then be defined to favor long lines. The scores from the ranking rule could then be averaged using region-perimeter-percentage as a weighting factor to form a simple shape measure.

Intersection rules are also applicable to the regions which a line intersects. In this format a score is calculated for the line based on the properties of the regions which it intersects. Having this flexibility introduces the problem of organising the application and results of the intersection rules in a meaningful way. The next section outlines the organisation method used in the current implementation along with some alternative organisation methods.

III.5. STRUCTURE AND SCORING OF COMPLEX AGGREGATIONS

There are many ways to apply relational rules to form complex aggregations. The method used in the current implementation of the VISIONS system is to define a primary element type through which other elements are grouped and a score formed. Other approaches include independent use of each element type, a hybrid approach which allows both of the above approaches to be used simultaneously, and an approach which allows previously formed aggregations to guide future element grouping. Each of these approaches has advantages and disadvantages which are outlined below.

The approach used in the current implementation is to define one of the element types, specifically regions, as the primary element type. All relational and object hypothesis scores are attached to the primary elements. Groups of secondary elements are only considered as a result of their intersection with the primary elements.

An intersection rule is used to support an initial object hypothesis by making it one component of a complex rule, as described in Section II.3. If a rule were written for the boundary lines of a region it would be added to the complex rule at the level of the component rules. In this case, the top-level complex rule would consist of the five region-based component rules and as many line-based rules as are needed to define the object. All of these component rules are combined uniformly by a weighted average. A top-level rule including line-based relational rules is shown in Figure 7.

The idea of defining a primary element type through which all grouping and scoring takes place can be applied

to any element type used. Regions were chosen over lines in the current implementation for two reasons. The first is that a single region is a better descriptor of image events than a single line. The second reason is that regions were the existing element type before line information was integrated into the system. By using regions the high-level code did not have to be changed.

A disadvantage of the existing implementation is that the representation of the complex aggregation and score is structured at the top level through elements in one segmentation representation, in this case region elements. It is possible that the line information could contain a perfect representation of an object, but that the region information is fragmented and unreliable. In this case a high score for the object will not be obtained since each sub-region of the object is intersected by only a subset of the lines for the object.

Figure 8 shows an example of this. The roof is fragmented into many smaller regions, but the line information shows the outline of the roof fairly clearly. In this case no region will get full benefit of the bounding lines since each region is only bound by a subset of the roof lines. The set of bounding lines could be extracted first, then the set of regions bound by these lines could be used to form an aggregate of regions on which the region score could be based.

One solution to this problem is to allow independent use of the information from each segmentation process. This method would apply rules to each type of segmentation element. Elements from each segmentation type with high scores on related rules could be combined if they intersect. The score of the aggregate of elements would then receive a higher score than each of the individual elements, reflecting the additional support for that object hypothesis. Figure 9 shows the structure of a complex rule which uses this method.

Using each type of segmentation information independently has the advantage of allowing good segmentation results to influence an object hypothesis score, even if the other segmentation processes did not extract the object. One disadvantage of this method is that some segmentation elements, lines for example, may not contain enough information by themselves to form a reliable object hypothesis. These types of elements need to be grouped before object hypothesis rules are applied. In the first aggregation method, where a primary element type is defined, secondary element type rules are applied only to the elements which are related to a primary element. Independent application of all rules is inefficient since each rule must be applied to every element in the segmentation.

A hybrid approach to the integration of segmentation information could be defined by allowing a combination of the above methods to be used. In some cases, areas of constant texture for instance, it is desirable to use regions as the primary element type. In other cases, particularly when the object is defined by a simple geometrical shape, lines should be used as the primary element type. Each object could be defined by a separate model for each type of segmentation information. These models could be rank-ordered based on the reliability of each type of segmentation process at extracting the primitive descriptors of the object being modeled. When these models are subsequently used the highest ranked model is considered the primary model. The other model types are used only on the segmentation elements which intersect the primary elements with good matches to the primary model. Only if the primary model fails to locate good matches would the secondary models be applied to the entire segmentation.

The hybrid approach has all the advantages of the independent approach. The difference is that only one type of object rule is applied to the entire image so the hybrid approach is more efficient. Also, the primary element type guides the grouping of secondary elements which avoids a combinatorial explosion.

A final approach solves these problems and also addresses the problem of forming aggregates of elements from one segmentation process. This approach uses the hybrid method mentioned above and, in addition, allows existing aggregations to guide the formation of additional aggregations.

In this approach existing aggregations which have missing elements, such as one side of a rectangle, are used to group additional elements. Using the rectangle example, the regions within the existing sides of the rectangle are grouped to form a new aggregation. This aggregation is then used in two ways. First, it is used as a single region entity to re-compute the region based rule scores. Second, it is used to group additional lines in hopes of finding the fourth side of the rectangle. In this way existing aggregations are used to form new aggregations, both of elements from one segmentation and elements from more than one segmentation. This approach reduces the number of combinations of elements considered for an aggregation, which in turn reduces the combinatorics of the problem and also provides a guide for grouping elements from one segmentation process.

IV. RESULTS

This section presents the results of running the intersection rules on urban house scenes and on road scenes. The results fall into two categories, those that derive texture measures and those that derive shape measures. Where appropriate the results of each component of a rule are presented to illustrate the advantages of integrating line and region information.

IV.1. RELATIONAL RULES USED FOR TEXTURE MEASUREMENT

A simple texture measure can be computed by counting the occurrences of short, high-contrast lines within a region and normalising by the region size. The filtering rule to compute this measure would use interior-line-percentage to select only those lines which are completely interior to the region. The ranking rule selects short, high-contrast lines. The combination function counts the number of lines with scores above some threshold and divides this number

by the size of the region in pixels. Figure 10 shows an example of this rule applied to a house scene.

Some objects, notably the roof of a house, are characterised by short horizontal lines interior to the region. A rule to produce high scores for these regions can be obtained by first writing a rule which selects short, horizontal lines. An intersection rule is then constructed from this line rule, a filter on interior-line-percentage which selects only interior lines, and a combination function which computes the density of lines whose contrast exceeds some threshold. The results of applying such a rule to a house scene are shown in Figure 11.

IV.2. RELATIONAL RULES USED FOR SHAPE ANALYSIS

A simple shape measure can be computed by determining how much of a region boundary is made up of long straight lines. The filtering rule for this measure uses interior-line-percentage to select only those lines which lie on a region boundary. The ranking rule is written to rank long lines higher than short lines. The combination function then uses region-perimeter-percentage to compute the percentage of a region boundary made up of lines which received a high score from the ranking rule. As Figure 12 shows this rule is useful for extracting roads.

As an extension to the previous rule, the long lines can also be ranked on orientation. An intersection rule of this type can be used to find regions with straight edges oriented in a certain direction. To define such a rule to find vertically oriented regions the ranking rule selects lines which are both long and vertical. The combination function then uses the scores of the ranking rule and the height of the region to determine if a region's vertical edges are straight. A rule of this type is applied to a house scene in Figure 13.

V. FUTURE WORK

The first, and easiest, extension to the existing system is to add additional segmentation processes to the low-level system. The types of information that could be added include motion, depth, and surface segmentations. Each segmentation process would create a set of elements with associated attributes which would be added to the intermediate-level representation. These new elements could then be used in the same way as regions and lines are used now. This would not involve any major modifications to the system, but would greatly enhance the reliability of its results.

Extensions considered for the relational measures include additional relational measures based on line support sets, the forming of artificial line support regions, and the use of chamfering for relational measures. Each of these modifications is aimed at increasing the accuracy and utility of relations between existing elements. It is assumed that when new types of elements (surfaces) are added to the representation, only those rules relating the new element type to existing types will have to be added to the rule set.

One extension being considered for the formation of complex aggregations is to use aggregations of lines instead of individual lines as the line entities. The aggregates of lines would be formed by grouping all lines with similar orientation or locality of endpoints [e.g., WEI85]. Attributes, such as shape features, could then be computed for each aggregation of lines. These entities, along with their attributes, would then be treated as new primitive elements by the system. The existing relational measures would still be meaningful for these entities since they are still pixel based. Additional measures would probably be desired, however, since the groups of lines would be complex structures.

The existing method of forming complex aggregations was implemented because of the ease with which it could be integrated with the existing system. As some of the problems of information fusion become better understood it is assumed that this method will be augmented with some of the ideas from Section III.5.

VI. CONCLUSION

The method described is a uniform, straight-forward way of combining evidence from multiple segmentation processes. Some problems exist with the region-based system, but the existing method significantly increases the reliability of initial object hypothesis scores for some objects.

VII. BIBLIOGRAPHY

[BAJ76] R. Bajcsy and M. Tavakoli, "Computer Recognition of Roads from Satellite Pictures," *IEEE Transactions on Systems, Man, and Cybernetics* SMC-6 (September 1976), pp. 623–637.

[BAR78] H. G. Barrow, J. M. Tennenbaum, R. C. Bolles, and H. C. Wolf "Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching," *Proc. DARPA IU Workshop* (May 1978), pp. 21–27.

[BEL85] R. L. Belknap, "An Interactive Rule Editing System for the Definition of Initial Object Hypothesis Rules," forthcoming COINS Technical Report, University of Massachusetts at Amherst, 1985.

[BUR84] J. B. Burns, A. R. Hanson, and E. M. Riseman, "Extracting Linear Features," *Proc. of the Seventh International Conference on Pattern Recognition* (July 30 – August 2, 1984), Montreal, Canada (to be published in PAMI).

[HAN78a] A. R. Hanson and E. M. Riseman, "VISIONS: A Computer System for Interpreting Scenes," *Computer Vision Systems* (A. Hanson and E. Riseman, eds.) (1978), pp. 303–333, Academic Press.

[HAN78b] A. R. Hanson and E. M. Riseman (Eds.), *Computer Vision Systems*, New York, Academic Press, 1978.

[HAN85] A. R. Hanson, E. M. Riseman, J. S. Griffith and T. E. Weymouth, "A Methodology for the Development of General Knowledge-Based Vision Systems," *Proc. IEEE Workshop on Principles of Knowledge-*

Based Systems, Denver, CO, December 1984, pp. 159–170.

[KOH83] R. R. Kohler. "Integrating Non-Semantic Knowledge into Image Segmentation Processes," Ph.D. Thesis, University of Massachusetts at Amherst, September 1983.

[KOH84] R. R. Kohler, "Integrating Non-Semantic Knowledge into Image Segmentation Processes," COINS Technical Report 84-04, University of Massachusetts at Amherst, March 1984.

[LAW85] D. T. Lawton, Personal Communication.

[MCK84] D. M. McKeown, W. A. Harvey and J. McDermott, "Rule Based Interpretation of Aerial Imagery," Dept. of Computer Science, Carnegie-Mellon University, (September 1984).

[NAG79] P. A. Nagin, "Studies in Image Segmentation Algorithms Based on Histogram Clustering and Relaxation," COINS Technical Report 79-15, University of Massachusetts at Amherst, September 1979.

[NAG82] P. A. Nagin, A. R. Hanson, and E. M. Riseman, "Studies in Global and Local Histogram-Guided Relaxation Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 3 (May 1982), pp. 263-277.

[NAS83] A. M. Nasif and M. D. Levine, "Low Level Segmentation: An Expert System," Technical Report TR-83-4 (April 1983), Electrical Engineering, McGill University.

[REY84] G. Reynolds, N. Irwin, A. R. Hanson, E. M. Riseman, "Hierarchical Knowledge-Directed Object Extraction Using a Combined Region and Line Representation," *IEEE Proceedings of the Workshop on Computer Vision: Representation and Control* (1984), pp. 238–247.

[RIS84] E. M. Riseman and A. R. Hanson, "A Methodology for the Development of General Knowledge-Based Vision Systems," *IEEE Proc. of the Workshop on Computer Vision: Representation and Control* (1984), pp. 159–170.

[SOU85] K. Southwick, "FEATSYS - An Intermediate-Level Representation of Image Feature Data," forthcoming Master's Thesis (expected February 1986), Computer and Infomation Science Department, University of Massachusetts at Amherst.

[WEI85] R. Weiss, A. R. Hanson, and E. M. Riseman, "Geometric Grouping of Straight Lines," *this proceedings*.

[WEY83] T. Weymouth, J. Griffith, A. R. Hanson, and E. M. Riseman, "Rule Based Strategies for Image Interpretation," *Proc. AAAI-83* (August 1983).

[WIL81] T. Williams, "Computer Interpretation of a Dynamic Image from a Moving Vehicle," Ph.D. Thesis and COINS Technical Report 81-22, University of Massachusetts at Amherst (May 1981).

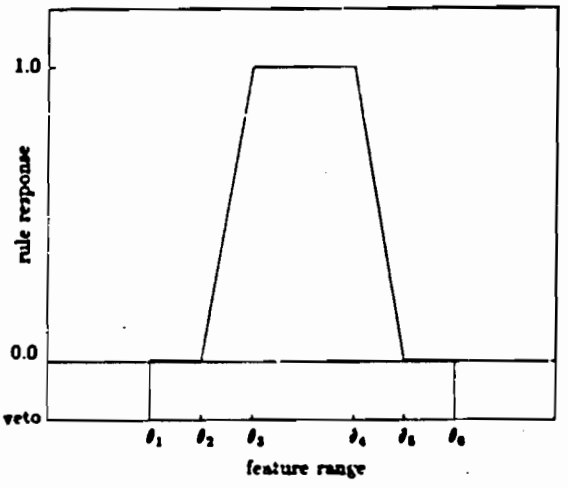


Figure 1. Structure of a simple rule for mapping an image feature measurement f_i into support for a label hypothesis on the basis of a prototype feature value. The object specific mapping is parameterised by six values, $\theta_1, \ldots, \theta_6$.

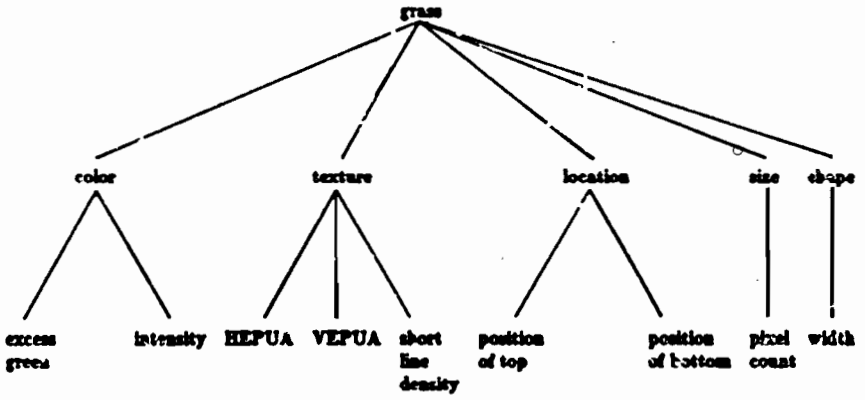


Figure 2. The structure of a complex rule for grass showing the five component rules based on region attributes.

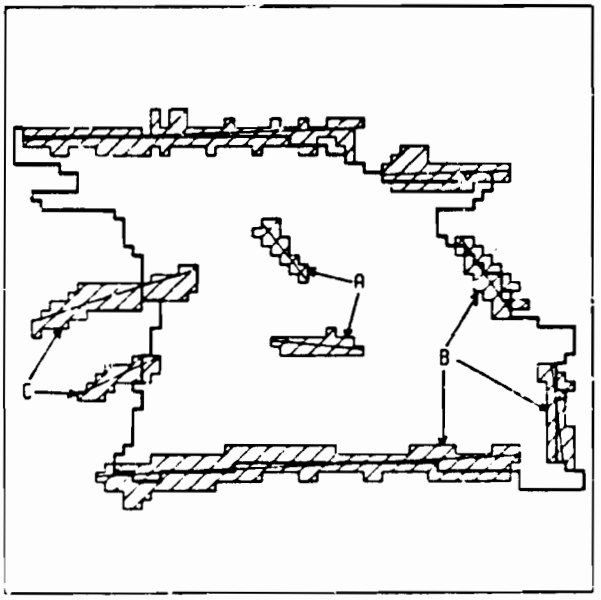


Figure 3. An intensity region with line suport regions and lines superimposed (line support regions are shaded). The three line catagories are labeled: A - interior, B - boundary, C - Superfluous.

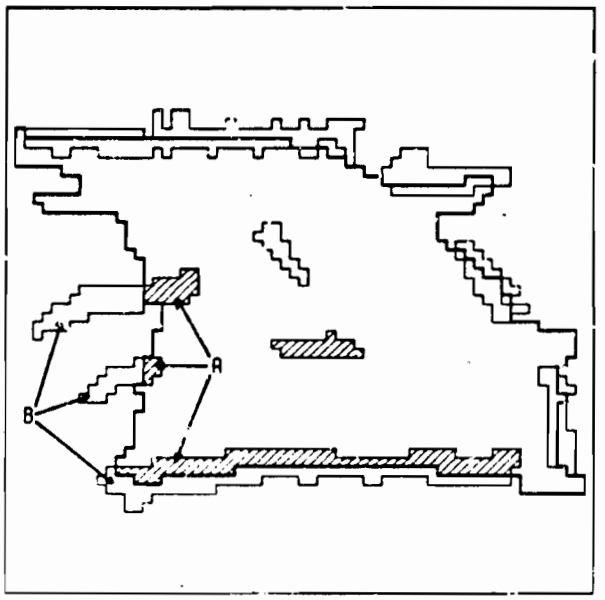


Figure 4. The derivation of interior-line-percentage. The measure is computed by dividing the number of pixels which intersect the region (A) by the total number of pixels in the line support set (E).

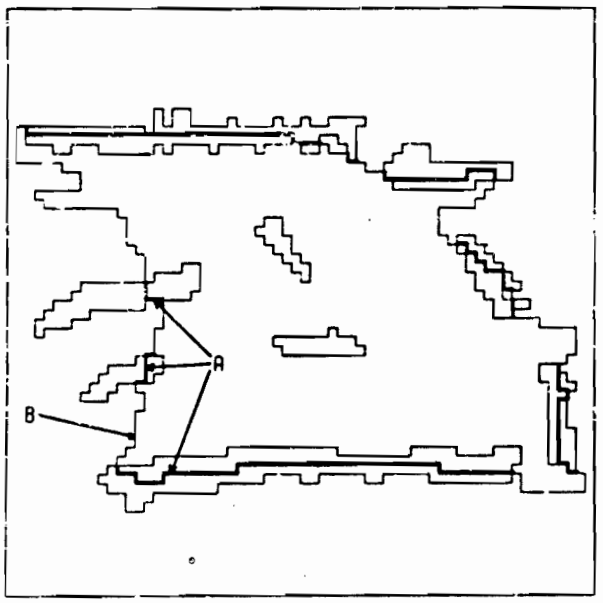


Figure 5. The derivation of region-perimeter-percentage. The measure is computed by dividing the length of region boundary covered by a line support set (A) by the length of the entire region boundary (B).

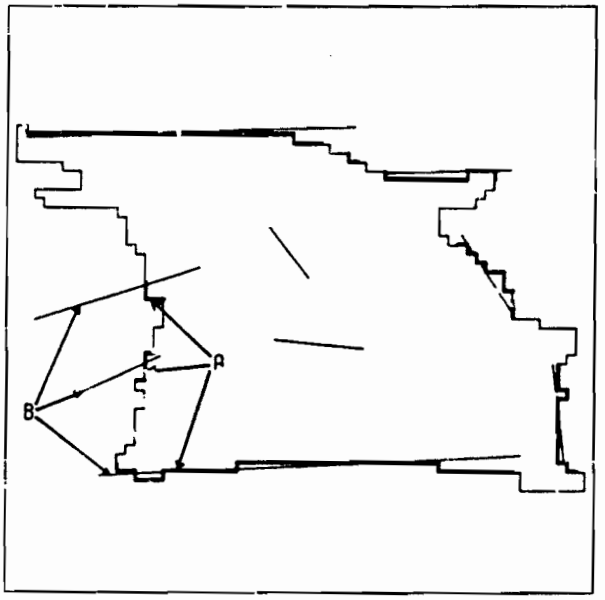


Figure 6. The derivation of line-length-percentage. The measure is computed by dividing the length of region boundary covered by a line support set (A) by the length of the line (B).

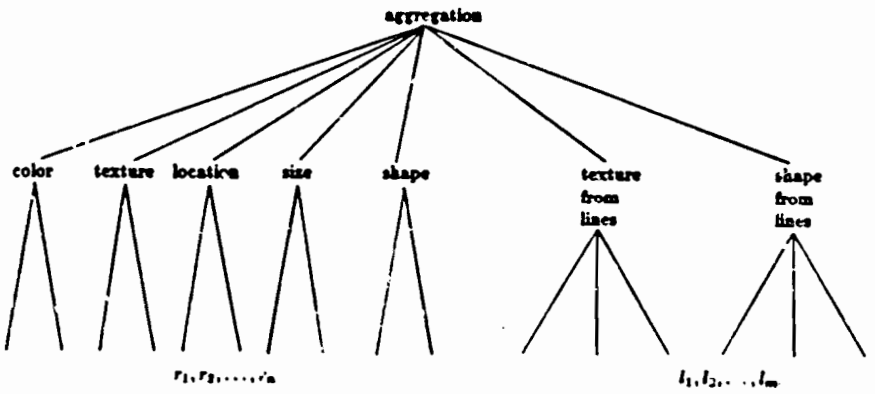


Figure 7. The structure of a complex rule with relational rules added for texture and shape. The rules $r_1, \ldots, r_n$ and $l_1, \ldots, l_m$ are rules over region and line attributes, respectively.

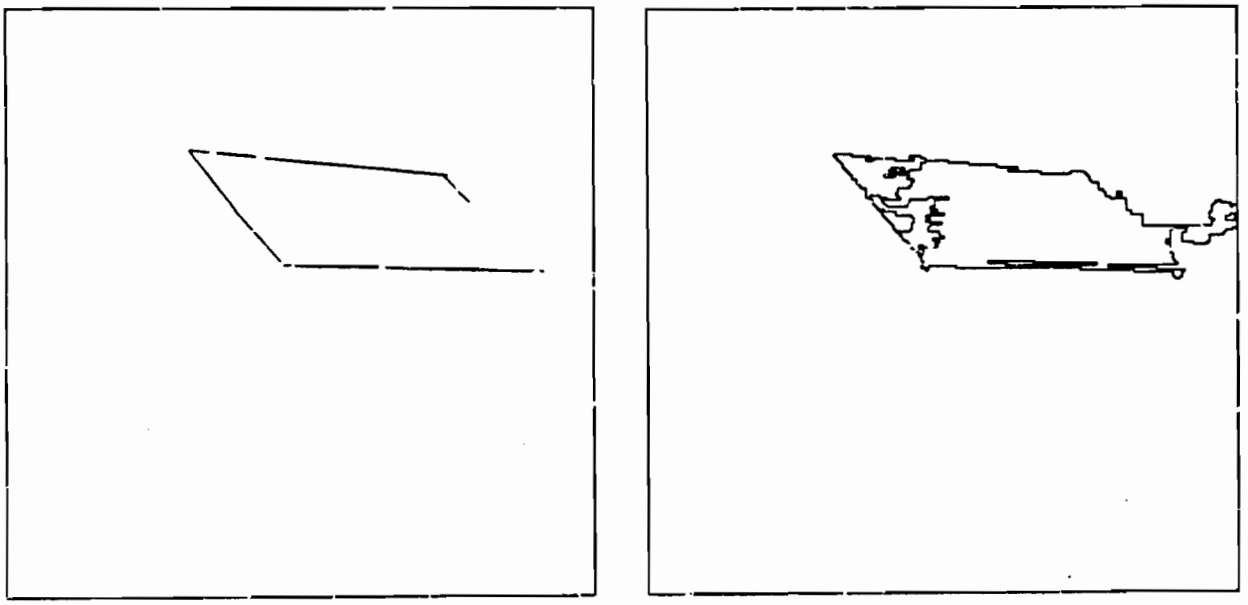


Figure 8. The lines bounding a roof (left) form a parallelogram, but the regions are fragmented (right) so intersection rules will not consider the entire set of lines for any one region.

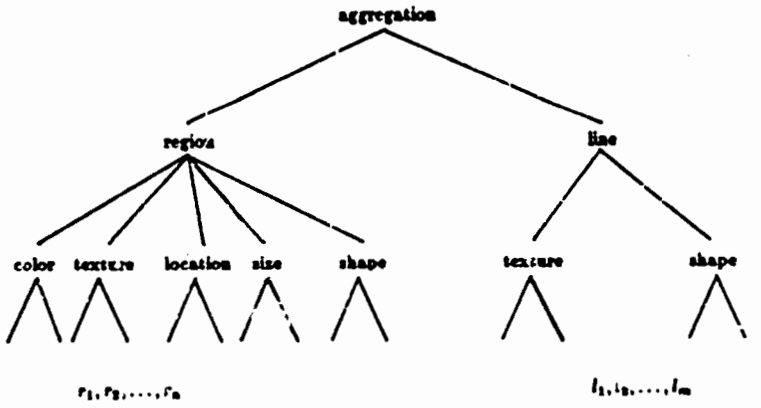


Figure 9. The structure of a rule for a complex aggregation formed by using line and region information independently. The rules $r_1, \ldots, r_n$ and $l_1, \ldots, l_m$ are rules over region and line attributes, respectively.

Figure 10. A simple texture measure computed by a relational rule which counts the density of lines within a region. The lines which received a high score from the filtering rule (lines interior to a region) are shown on the left. On the right, rule scores are shown mapped back to the regions. Densely shaded regions received high scores.

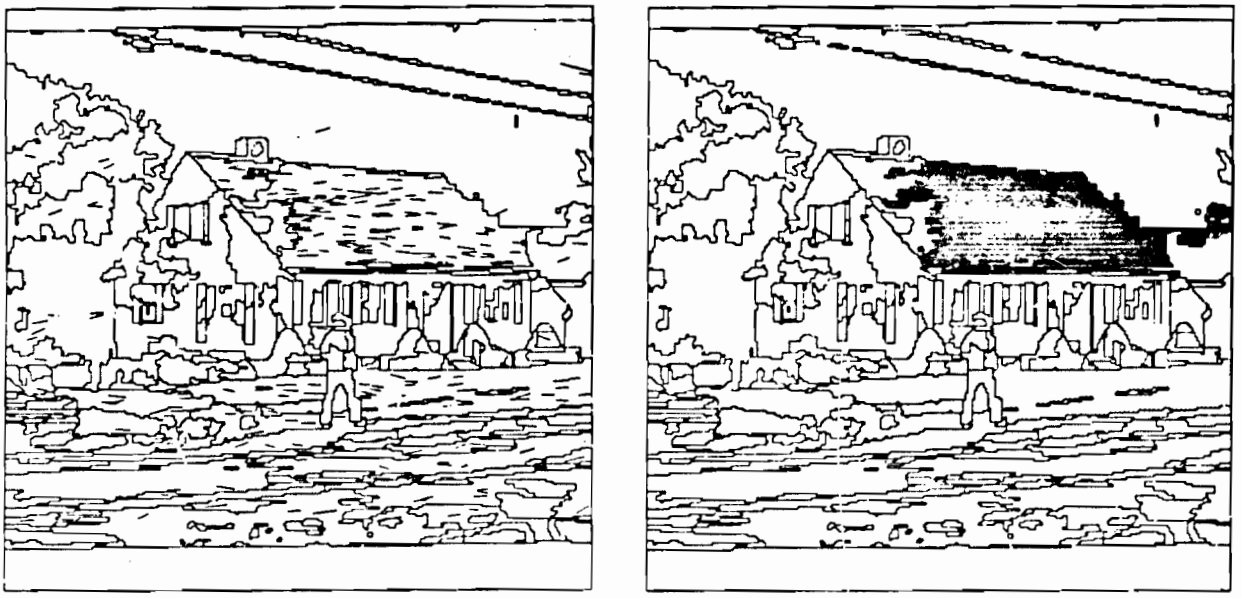


Figure 11. A slightly more complex texture measure in which the orientation of the lines is taken into account. On the left are the lines which received a high score from the ranking rule and the filtering rule (short, horizontal lines which are interior to a region). The region scores are shown on the right.

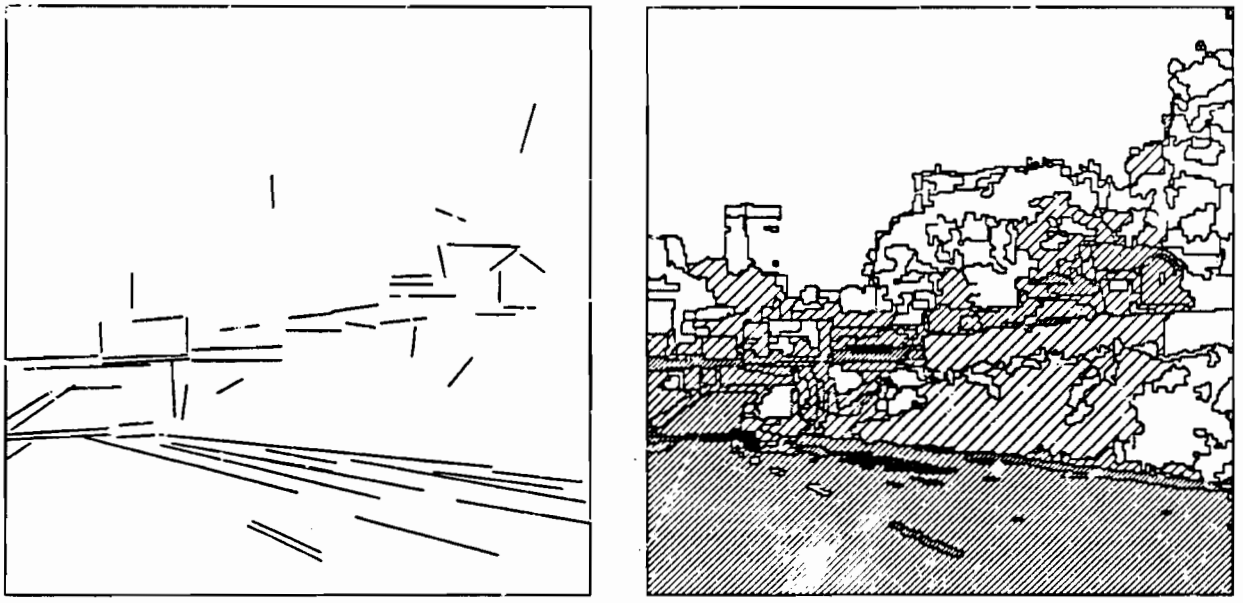


Figure 12. A relational rule to find regions which are bounded by long lines. On the left are the lines which received high scores from the filtering and ranking rules (long lines which lie on a region boundary). The region scores are mapped back to the image on the right.

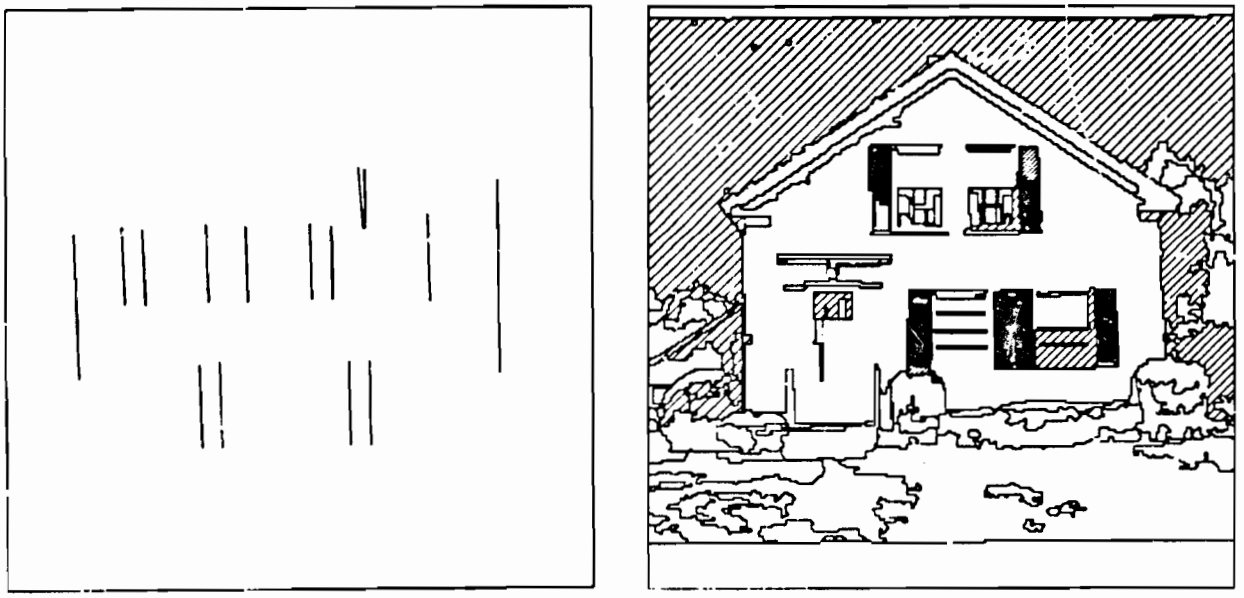


Figure 13. A simple shape measure computed by a relational rule which measures the straightness of vertical region boundaries. The lines which received a high score from the filtering rule and the ranking rule (long, vertical lines which lie on a region boundary) are shown on the left. On the right, rule scores are shown mapped back to the regions.

SECTION IV

TECHNICAL PAPERS NOT PRESENTED

GC970145-34

PROBABILISTIC SOLUTION OF ILL-POSED PROBLEMS IN COMPUTATIONAL VISION

J. Marroquin[1], S. Mitter[2], and T. Poggio

The Artificial Intelligence Laboratory, Massachusetts Institute of Technology
[1]Artificial Intelligence Laboratory and Laboratory for Information and Decision Systems
[2]Laboratory for Information and Decision Systems

Computational vision is a set of inverse problems. We review standard regularization theory, discuss its limitations, and present new stochastic (in particular, Bayesian) methods for their solution. We derive efficient algorithms and describe parallel implementations on digital parallel SIMD architectures, as well as a new class of parallel hybrid computers.

1. Introduction

1.1 Computational Vision

Computational vision denotes a new field in artificial intelligence that has developed in the last 15 years. Its two main goals are to develop image understanding systems which automatically could provide scene descriptions from real images, and to understand biological vision. Its main focus is on theoretical studies of vision, considered as an information processing task.

Since at least the work of David Marr (Marr, 1982; see also Marr & Poggio, 1977), it has been customary to consider vision as an information processing system that could be divided into several modules at different theoretical levels, at least as a first approximation. In particular, Marr suggested that the goal of the first step of vision is to obtain descriptions of physical properties of three-dimensional surfaces around the viewer such as distance, orientation, texture, and reflectance. This first step of vision, up to what has been called *2-1/2 D sketch* or *intrinsic images*, is mainly bottom-up relying on general knowledge but no special high-level information about the scene to be analyzed.

The first part of vision–from images to surfaces–has been termed *early vision*. Although this point-of-view has been embraced widely (see a set of recent reviews, e.g., Brown, 1984; Brady, 1981; Barrow & Tannenbaum, 1981; Poggio, 1984), it is important to observe that its correctness is still to be proven. In particular, it is still unclear what the nature of the 2-1/2-D sketch representation is, how different visual modules interact, how their output is fused and what is the role of high-level knowledge on early visual processes. The critical problem of the organization of vision and of the control of the flow of information from the different modules and how high-level knowledge is used is still very much an open problem.

In this paper, we do not consider this larger issue. Our point-of-view is that a rigorous analysis of individual modules of vision is bound to play an important role in any full theory of vision.

1.2. Early Vision

Early vision consists of a set of processes that recover physical properties of visible three-dimensional surfaces from the two-dimensional images. Computational, biological and epistemological arguments (see Marr and Poggio, 1977) suggest that early vision processes are generic ones that correspond to conceptually independent modules that can be studied, at least to a very first approximation, in isolation. Table 1 shows a list of some of the early vision modules.

Table 1

Examples of early vision processes

- **Edge detection**
- **Spatio–temporal interpolation and approximation**
- **Computation of optical flow**
- **Computation of lightness and albedo**
- **Shape from contours**
- **Shape from texture**
- **Shape from shading**
- **Binocular stereo matching**
- **Structure from motion**
- **Structure from stereo**
- **Surface reconstruction**
- **Computation of surface colour**

The standard definition of computational vision is that it is inverse optics. The direct problem–the problem of classical optics–or computer graphics–is to determine the images of three-dimensional objects. Computational vision is confronted with inverse problems of recovering surfaces from images. Much infor-

mation is lost during the imaging process that projects a three-dimensional world into two-dimensional arrays (images). As a consequence, vision must rely on natural constraints, that is, general assumptions about the physical world to derive an unambiguous output. This is typical of many inverse problems in mathematics and physics.

In fact, the common characteristics of most early vision problems, in a sense their deep structure, can be formalized: *early vision problems are ill-posed in the sense defined by Hadamard.* A problem is well-posed when its solution (a) exists, (b) is unique and (c) depends continuously on the initial data. Ill-posed problems fail to satisfy one or more of these criteria.

Bertero, Poggio and Torre (1986) show precisely the mathematically ill-posed structure of several problems listed in Table 1 (see also Poggio and Torre, 1984.) The recognition that early vision problems are ill-posed suggests immediately the use of regularization methods developed in mathematics and mathematical physics for solving the ill-posed problems of early vision (Poggio & Torre, 1984).

1.3. Standard Regularization in Early Vision

The main idea for "solving" ill-posed problems is to restrict the class of admissible solutions by introducing suitable *a priori* knowledge. In standard regularization methods, due mainly to Tikhonov, the regularization of the ill-posed problem of finding z from the data y:

$$Az = y$$

requires the choice of norms $\|\cdot\|$ and of a stabilizing functional $\|Pz\|$. In standard regularization theory, A is a linear operator, the norms are quadratic and P is linear. A method that can be applied is:

Find z that minimizes

$$\|Az - y\|^2 + \lambda\|Pz\|^2, \qquad (1)$$

where λ is a so-called regularization parameter.

In this method, λ controls the compromise between the degree of regularization of a solution and its closeness to the data (the first term in equation 1). P embeds the physical constraints of the problem. It can be shown for quadratic variational principles that under mild conditions the solution space is convex and a unique solution exists.

Poggio et al (1984, 1985) show that several problems in early vision can be "solved" by standard regularization techniques. Surface reconstruction, optical flow at each point in the image, optical flow along contours, color, stereo can be computed by using standard regularization techniques. Variational principles that are not exactly quadratic but have the same form as equation 1 can be used for other problems in early vision. The main results of Tikhonov can, in fact, be extended to some cases in which the operators A and P are nonlinear, provided they satisfy certain conditions. (Morozov, 1984.)

Standard regularization methods can be implemented very efficiently by parallel architectures of the fine-grain type, such as the Connection Machine (Hillis, 1985). Analog networks, either electrical or chemical, can also be a natural way of solving the variational principles dictated by standard regularization theory (Poggio & Koch, 1984, 1985). A list of the problems that can be regularized by standard regularization theory or slightly non-linear versions of it are listed in Table 2, together with the associated regularization principle.

1.4. Limitations of Standard Regularization Theory

This new theoretical framework for early vision shows clearly not only the attractions, but also the limitations that are intrinsic to the standard Tikhonov form of regularization theory. Standard regularization methods lead to satisfactory solutions of early vision problems but cannot deal effectively and directly with a few general problems such as *discontinuities* and *fusion of information from multiple modules.*

Standard regularization theory with linear A and P is equivalent to restricting the space of solution to generalized splines, whose order depends on the order of the stabilizer P. This means that in some cases the solution is too smooth, and cannot be faithful in locations where discontinuities are present. In optical flow, surface reconstruction and stereo, discontinuities are in fact not only present, but also the most critical locations for subsequent visual information processing. Standard regularization cannot deal well with another critical problem of vision, the problem of fusing information from different early vision modules. Since the regularizing principles of the standard theory are quadratic, they lead to linear Euler-Lagrange equations. The output of different modules can therefore be combined only in a linear way. Terzopoulous (1984; see also Poggio et al., 1985) has shown how standard regularization techniques can be used in the presence of discontinuities in the case of surface interpolation. After standard regularization, locations where the solution f originates a large error in the second term of equation 1, are identified (this needs setting a threshold for the error in smoothness). A second regularization step is then performed using the location of discontinuities as boundary conditions.

A similar method could be used for fusing information from multiple sources: a regularizing step could be performed and locations where terms of the type of the first term of equation (1) give large errors would be identified. A decision step would then follow by setting appropriately various controlling parameters in those

Table 2

Problem	Regularization Principle
Edge detection	$\int \left[(Sf - i)^2 + \lambda (f_{xx})^2\right] dx$
Optical Flow (area based)	$\int \left[(i_x u + i_y v + i_t)^2 + \lambda (u_x^2 + u_y^2 + v_x^2 + v_y^2)\right] dxdy$
Optical Flow (contour based)	$\int \left[(\mathbf{V} \cdot \mathbf{N} - V^N)^2 + \lambda (\frac{\partial}{\partial s} \mathbf{V})^2\right] ds$
Surface Reconstruction	$\int \left[(S \cdot f - d)^2 + \lambda (f_{xx}^2 + 2 f_{xy}^2 + f_{yy}^2)^2\right] dxdy$
Spatiotemporal approximation	$\int \left[(S \cdot f - i)^2 + \lambda (\nabla f \cdot \mathbf{V} + f_t)^2\right] dxdy$
Color	$\|I^* - Ax\|^2 + \lambda \|Px\|^2$
Shape from Shading	$\int \left[(E - R(f,g))^2 + \lambda (f_x^2 + f_y^2 + g_x^2 + g_y^2)\right] dxdy$
Stereo	$\int \left\{ \left[\nabla^2 G * \left(L(x,y) - R(x + d(x,y), y)\right)\right]^2 + \lambda (\nabla d)^2 \right\} dxdy$

locations, therefore weighting in an appropriate way (for instance, vetoing some of) the various contributing processes.

In any case one would like a more comprehensive and coherent theory capable of dealing directly with the problem of discontinuities and the problem of fusing information. So the challenge for a regularization theory of early vision is to extend it beyond standard regularization methods and their most obvious non-linear versions.

1.5. Stochastic Route to Regularizing Early Vision

In this paper, we will outline a rigorous approach to overcome part of the ill-posedness of vision problems, based on Bayes estimation and Markov Random Field models, that effectively deals with the problems faced by the standard regularization approach. In this approach, the *a priori* knowledge is represented in terms of an appropriate probability distribution, whereas in standard regularization *a priori* knowledge leads to restrictions on the solution space. This distribution, together with a probabilistic description of the noise that corrupts the observations, allows one to use Bayes theory to compute the posterior distribution $P_{f|g}$, which represents the likelihod of a solution f given the observations g. In this way, we can solve the reconstruction problem by finding the estimate $\hat{f}$ which either maximizes this a posteriori probability distribution (the so called Maximum a Posteriori or MAP estimate), or minimizes the expected value (with respect to $P_{f|g}$) of an appropriate error function. The class of solutions that can be obtained in this way is much larger than in standard regularization. In particular, we will show under which conditions this new method leads to solutions that are of the standard regularization type (see section 3).

The price to be paid for this increased flexibility is computational complexity. New parallel architectures and possibly hybrid computers of the digital-analog type promise however to deal effectively with the computational requirements of the methods proposed here. We will discuss at the end of the paper in some detail these new parallel architectures.

2. Probabilistic Models

The key to the success in the use of this approach, is our ability to find a class of stochastic models (that is, random fields) that have the following characteristics:

(i) The probabilistic dependencies between the elements of the field should be local. This condition is necessary if the field is to be used to model surfaces that are only piecewise smooth; besides, if it is satisfied, the reconstruction algorithms are likely to be distributed, and thus, efficiently implementable in parallel

hardware.

(ii) The class should be rich enough, so that a wide variety of qualitatively different behaviors can be modeled.

(iii) The relation between the parameters of the models and the characteristics of the corresponding sample fields should be relatively transparent, so that the models are easy to specify.

(iv) It should be possible to represent the prior probability distribution P_f explicitly, so that Bayes theory can be applied.

(v) It should be possible to specify efficient Monte Carlo procedures, both for generating sample fields from the distribution, so that the capability of the model to represent our prior knowledge can be verified, and to compute the optimal estimators.

A class of random fields that satisfies these requirements is the class of Markov Random Fields (MRF's) on finite lattices (see Wong, 1968 and Woods, 1972). A MRF has the property that the probability distribution of the configurations of the field can always be expressed in the form of a Gibbs distribution:

$$P_f(f) = \frac{1}{Z} e^{-\frac{1}{T_0} U(f)}$$

where Z is a normalizing constant, T_0 is a parameter (known as the "natural temperature" of the field) and the "Energy function" $U(f)$ is of the form:

$$U(f) = \sum_C V_C(f)$$

where C ranges over the "cliques" associated with the neighborhood system of the field, and the potentials $V_C(f)$ are functions supported on them (a clique is either a single site, or a set of sites such that any two sites belonging to it are neighbors of each other).

As an example, the behavior of piecewise constant functions can be modeled using first order MRF models on a finite lattice L with generalized Ising potentials (Geman and Geman, 1984):

$$V_C(f_i, f_j) = \begin{cases} -1, & \text{if } |i-j| = 1 \text{ and } f_i = f_j \\ 1, & \text{if } |i-j| = 1 \text{ and } f_i \neq f_j \\ 0, & \text{otherwise} \end{cases}$$

$$f_i \in Q_i = \{q_1, \ldots, q_M\} \quad \text{for all } i \in L$$

We will use a free boundary model, so that the neighborhood size for a given site will be: 4, if it is in the interior of the lattice; 3, if it lies at a boundary, but not at a corner, and 2 for the corners.

The Gibbs distribution:

$$P_f(f) = \frac{1}{Z} \exp[-\frac{1}{T_0} U_0(f)]$$

$$U_0(f) = \sum_{i,j} V(f_i, f_j) \tag{2}$$

defines a one parameter family of models (indexed by T_0) describing piecewise constant patterns with varying degrees of granularity.

We will assume that the available observations g are obtained from a typical realization f of the field by a degrading operation (such as sampling) followed by corruption with i.i.d. noise (the form of whose distribution is known), so that the conditional distribution can be written as:

$$P_{g|f}(g; f) = \exp[-\alpha \sum_{i \in S} \Phi_i(f, g_i)] \tag{3}$$

where $\{\Phi_i\}$ are some known functions, and α is a parameter.

The posterior distribution is obtained from Bayes rule:

$$P_{f|g}(f; g) = \frac{1}{Z_P} \exp[-U_P(f; g)] \tag{4}$$

with

$$U_P(f; g) = \frac{1}{T_0} U_0(f) + \sum_{i \in S} \Phi(f, g_i) \tag{5}$$

For example, in the case of binary fields ($M = 2$) with the observations taken as the output of a binary symmetric channel (BSC) with error rate ϵ (Gallager, 1975), we have:

$$P(g_i \mid f_i) = \begin{cases} (1-\epsilon), & \text{for } g_i = f_i \\ \epsilon, & \text{for } g_i \neq f_i \end{cases}$$

The posterior energy reduces to:

$$U_P(f; g) = \frac{1}{T_0} \sum_{i,j} V(f_i, f_j) + \alpha \sum_i (1 - \delta(f_i - g_i)) \tag{6}$$

where $f_i \in \{q_1, q_2\}$;

$$\delta(a) = \begin{cases} 1, & \text{if } a = 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

and

$$\alpha = \ln\left(\frac{1-\epsilon}{\epsilon}\right) \tag{8}$$

3. Cost Functionals

The Bayesian approach to the solution of reconstruction problems has been adopted by several researchers. In most cases, the criterion for selecting the optimal estimate has been the maximization of the posterior probability (the Maximum a Posteriori or MAP estimate). It has been used, for example, by Geman and Geman (1984) for the restoration of piecewise constant images; by Grenander (1984) for pattern reconstruction, and by Elliot et. al. (1983) and Hansen and Elliot (1982) for the

segmentation of textured images (a similar criterion — the maximization of a suitably defined likelihood function — has been used by Cohen and Cooper (1984) for the same purposes).

In some other cases, a performance criterion, such as the minimization of the mean squared error has been implicitly used for the estimation of particular classes of fields. For example, for continuous-valued fields with exponential autocorrelation functions, corrupted by additive white Gaussian noise, Mani and Assefi (1972) and Habibi (1972) have used causal linear models and optimal (Kalman) linear filters for solving the reconstruction problem.

The minimization of the expected value of error functionals, however, has not been used as an explicit criterion for designing optimal estimators in the general case. We will show that this design criterion is in fact more appropriate in our case, for the following reasons:

(i) It permits one to adapt the estimator to each particular problem.

(ii) It is in closer agreement with one's intuitive assessment of the performance of an estimator.

(iii) It leads to attractive computational schemes.

As an example, we will now propose design criteria for two particular problems: image segmentation and surface reconstruction.

Consider a field f with N elements each of which can belong to one of a finite set Q_i of classes. Let f_i denote the class to which the i^{th} element belongs. The segmentation problem is to estimate f from a set of observations $\{g_1, \ldots, g_n\}$. Note that f_i does not necessarily correspond to the image intensity. It may represent, for example, the texture class for a region in the image (as in Elliot et. al., 1983), etc.

A reasonable criterion for the performance of an estimate $\hat{f}$ is the number of elements that are not classified correctly. Therefore, we define the segmentation error e_s as:

$$e_s(f, \hat{f}) = \sum_{i=1}^{N} (1 - \delta(f_i - \hat{f}_i)) \qquad , f_i, \hat{f}_i \in Q_i \qquad (9)$$

In the case of the reconstruction problem, an estimate $\hat{f}$ should be considered "good" if it is close to f in the ordinary sense, so that the total squared error:

$$e_r(f, \hat{f}) = \sum_{i=1}^{N} (f_i - \hat{f}_i)^2 \qquad (10)$$

will be a reasonable measure for its performance.

To derive the optimal estimators with respect to the criteria stated above, we first present the general result (which can be found, for example in Abend, 1968) which states that if the posterior marginal distributions for every element of the field are known, the optimal Bayesian estimator with respect to any additive, positive definite cost functional C may be found by independently minimizing the marginal expected cost for each element.

In more precise terms, we will consider cost functionals $C(f, \hat{f})$ of the form:

$$C(f, \hat{f}) = \sum_{i \in L} C_i(f_i, \hat{f}_i)$$

with

$$C_i(a, b) \begin{cases} = 0, & \text{if } a = b \\ > 0, & \text{if } a \neq b \end{cases} \text{, for all } i$$

We will assume that the value of each element f_i of the field f is constrained to belong to some finite set Q_i (the generalization to the case of compact sets is straightforward). The Optimal Bayesian estimator $\hat{f}^*$ with respect to the cost functional C is defined as the global minimizer of the expected value of C over all possible f and g. One can prove that this estimate can be found by minimizing independently the marginal expected cost for each element, i.e.,

$$\hat{f}_i^* = q \quad . \quad \sum_{r \in Q_i} C_i(r, q) P_i(r \mid g) \leq \sum_{r \in Q_i} C_i(r, s) P_i(r \mid g)$$

$$\text{for all } s \neq q, \text{ and for all } i \in L$$

where $P_i(r \mid g)$ is the posterior marginal distribution of the element i:

$$P_i(r \mid g) = \sum_{f : f_i = r} P_{f|g}(f; g)$$

The optimal estimators for the error criteria defined above, can be easily derived from this result:

In the case of the segmentation problem, we get that

$$\hat{f}_i^* = q \in Q_i \quad : \quad P_i(q \mid g) \geq P_i(s \mid g)$$

$$\text{for all } s \neq q \qquad (11)$$

We will call this estimate the "Maximizer of the Posterior Marginals" ($\hat{f}_{MPM}$).

For the reconstruction problem, the optimal estimate is:

$$\hat{f}_i^* = q \in Q_i \quad : \quad (\hat{f}_i - q)^2 \leq (\hat{f}_i - s)^2$$

$$\text{for all } s \neq q \qquad (12)$$

We will call this estimate the "Thresholded Posterior Mean" ($\hat{f}_{TPM}$).

The main obstacle for the practical application of these results, lies in the formidable computational cost associated with the exact computation of the marginals and the mean of the posterior distribution given by (5), even for lattices of moderate size. In the next section we will present a general distributed procedure that will permit us to approximate these quantities as precisely as we may want.

4. Algorithms.

The algorithms that we will propose are based on the use of the Metropolis (Metropolis e. al., 1956) or Gibbs Sampler (Geman and Geman, 1984) schemes, to simulate the equilibrium behavior of the coupled MRF described by equation (5). We recall that the Markov chain generated by these algorithms is regular, and their invariant measure is the posterior distribution $P_{f|g}$. The law of large numbers for regular chains (see, for example, Kemeny and Snell, 1960) establishes that the fraction of time that the chain will spend on a given state f will tend to $P_{f|g}(f;g)$ as the number of steps gets large, independently of the initial state. This means that we can approximate the posterior marginals by:

$$P_i(q \mid g) \approx \frac{1}{k-n} \sum_{t=k}^{n} \delta(f_i^{(t)} - q) \qquad (13)$$

and $\bar{f}$ by:

$$\bar{f}_i \approx \frac{1}{n-k} \sum_{t=k}^{n} f_i^{(t)} \qquad (14)$$

where $f^{(t)}$ is the configuration generated by the Metropolis algorithm at time t, and k is the time required for the system to be in thermal equilibrium. From these values, $\hat{f}_{MPM}$ and $\hat{f}_{TPM}$ can be easily computed using (11) and (12).

This procedure is related to the use of simulated annealing for finding the global minimum of U_f (i.e., the MAP estimate: see Geman and Geman, 1984). In our case, however, we are interested in gathering statistics about the equilibrium behavior of the coupled field *at a fixed temperature* $T = 1$, rather than in finding the ground state of the system. This fact gives our procedure some distinct advantages:

1. It is difficult to determine in general the descent rate of the temperature (annealing schedule) that will guarantee the convergence of the annealing process in a reasonable time (it usually involves a trial and error procedure). Since we are running the Metropolis algorithm at a fixed temperature, this issue becomes irrelevant.

2. Since in our case we are using a Monte Carlo procedure to approximate the values of some integrals, we should expect a nice convergence behavior, in the sense that coarse approximations can be computed very rapidly, and then refined to an arbitrary precision (in fact, it can be proved (see Feller, 1950) that the expected value of the squared error of the estimates (13) and (14) is inversely proportional to n).

The main disadvantage of this procedure is that in the case of the segmentation problem, a large amount of memory might be required if the number of classes per element m is large (we need to store the $N(m-1)$ numbers that define the posterior marginals).

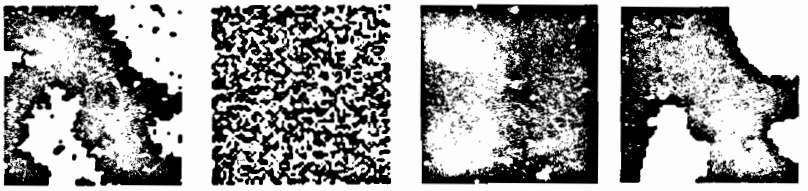
With respect to the relative performance, we point out that in many cases, particularly for high signal to noise ratios, the MAP estimate is usually close to the optimal one. If the noise level is high, however, the difference in the performances of the two estimators may be dramatic. This is illustrated in the example portrayed in figure 1: Panel (a) represents a typical realization of a 64×64 binary Ising net with free boundaries, using a value of $T_0 = 1.71$ (0.75 times the critical temperature of the lattice); panel (b), the output of a binary symmetric channel with error rate $\epsilon = 0.4$; panel (c) the MAP estimate, and panel (d) an approximation to the MPM estimate (which we will label "MPM (M.C.)") obtained using the Metropolis algorithm and equation (10) to estimate the posterior density. The corresponding values of the posterior energy U_f (equation (13)) and the relative segmentation error ($e_r/64^2$) are shown on Table 3.

It is clear that the approximation to the MPM estimates shown in panel (d) is better than the MAP from almost any viewpoint.

An intuitive explanation for this behavior comes from the fact that the MAP estimator is implicitly minimizing the expected value of a cost functional $C_{MAP}(f, \hat{f})$ which is equal to zero only if $f_i = \hat{f}_i$ *for all* i, and is equal to, say, M otherwise. If the signal to noise ratio is sufficiently high, the expected value of the optimal segmentation error will be very close to zero, so that $\hat{f}_{MPM}$ and $\hat{f}_{MAP}$ will coincide. In a high noise situation, however, the MAP estimator will tend to be too conservative, since from its viewpoint it is equally

Table 3

	f	g	$\hat{f}_{MAP}$	$\hat{f}_{MPM}(M.C.)$	$\hat{f}_{MPM}(Det.)$
Energy	−5594.8	−226.0	−6660.9	−6460.0	−6427.0
Seg. Error	–	0.4	0.33	0.128	0.124



(a) (b) (c) (d)

Figure 1. (a) Sample function of a binary MRF. (b) Output of a binary symmetric channel (error rate: 0.4) (c) MAP estimate. (d) Monte Carlo approximation to the MPM estimate.

costly to make one or one thousand mistakes. The MPM estimator, in contrast, can make a better (although more risky) guess, since making a few mistakes has only a marginal effect on the expected cost.

A quantitative comparison of the performances of the MAP and MPM estimators, with respect to the segmentation error, can be obtained using the ratio:

$$r = \frac{\bar{e}_{MAP}}{\bar{e}_{TPM}} =$$

$$= \frac{\sum_{f,g} \exp[-U_P(f;g)]e_s(f, \hat{f}_{MAP}(g))}{\sum_{f,g} \exp[-U_P(f;g)]e_s(f, \hat{f}_{TPM}(g))}$$

In figure 2 we show a plot of the ratio r for a 2×2 lattice, for different values of the error rate ϵ and the natural temperature T_0. As expected, r is never less than 1. In the worst case (for $\epsilon = 0.1$ and $T_0 = 0.2$) the error of the MAP estimate is 1.17 times that of the MPM estimate; if T_0 is not too small and ϵ is not too large, both estimates coincide, and as ϵ approaches 0.5 (low signal to noise ratio), the MPM estimate is consistently better than the MAP. An experimental analysis of larger lattices reveals a similar qualitative behavior, but the values of r are much larger in this case (see Table 3).

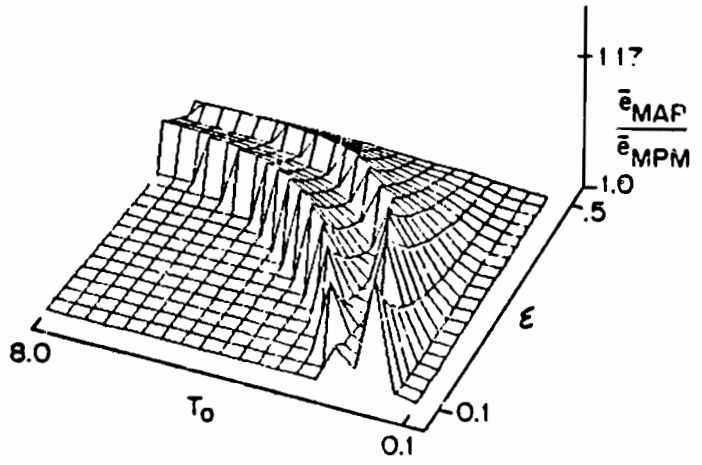
5. Examples of Applications in Vision

5.1. Recostruction of Piecewise Constant Functions

The efficient solution of this problem is relevant for several reasons: binary images (or images consisting of only a few grey levels) are directly useful in many interesting applications (for example, object recognition and manipulation in restricted (industrial) environments;, besides, several perceptual problems, such as the segmentation of textured images (Elliot, et. al. (1983); Hansen and Elliot (1982); Cohen and Cooper (1984)), or the formation of perceptual clusters (Marroquin, 1985) can be reduced to the problem of reconstructing a piecewise constant surface.

The prior model for this kind of functions is given by equations (1) and (2), and the posterior distribution, by equation (4). If the parameters that characterize the system (namely, the "natural temperature" T_0 and the noise parameter α) are known, the MPM estimator

Figure 2. Ratio of the average errors of the MAP and MPM estimators for a 2×2 Ising net.



produces excellent results, such as the one illustrated in figure 1.

In most practical cases, however, we are only given the noisy observations g and general qualitative information about the structure of the field and the noise, so that f, α (which stands, for example, for the error rate ϵ when the noise corruption corresponds to a BSC, or for the variance, σ^2, in the case of additive Gaussian noise) and T_0 have to be simultaneously estimated.

In principle, one could use again a Bayesian approach, and assuming prior independent uniform distributions for α and T_0 (in the ranges $[\alpha^0, \alpha^1]$ and $[T_0^0, T_0^1]$, respectively), find those $\hat{\alpha}, \hat{T}_0$ and $\hat{f}$ which jointly maximize the posterior distribution:

$$P(f, \alpha, T_0 \mid g) = \frac{\exp[-U_P(\alpha, T_0, f)]}{(\alpha^1 - \alpha^0)(T_0^1 - T_0^0) Z(T_0) P_g(g)}$$

The main difficulty here is the extraordinary computational complexity of the partition function:

$$Z(T_0) = \sum_f \exp[-\frac{1}{T_0} U_0(f)]$$

which makes this approach impractical, except for very small lattices.

Another approach, with which we have obtained very good results, consists on defining a merit function for an estimate (obtained using a particular value for the parameters), which is related to the degree of uniformity in the spatial distribution of the corresponding residuals. We have used, for example, a likelihood function L, which we obtain by covering the lattice with a set of m non-overlapping squares (say, 8 pixels wide); computing the relative variance of the noise parameter, estimated over each square, and adding all these terms together.

$$L(\hat{f}) = -\sum_{j=1}^{m} \left(\frac{\hat{\alpha} - \hat{\alpha}_j}{\hat{\alpha}} \right)^2$$

where $\hat{\alpha}$ and $\hat{\alpha}_j$ denote the conditional (on $\hat{f}$) maximum likelihood estimates of the noise parameter, obtained using the residuals over the whole lattice, and over the j^{th} square, respectively. The optimal estimate for f is then obtained as the global maximizer of L over the appropriate region of the parameter space. An example of the performance of this scheme is presented in figure 3, which shows the restoration of a ternary pattern corrupted by additive, white Gaussian noise.

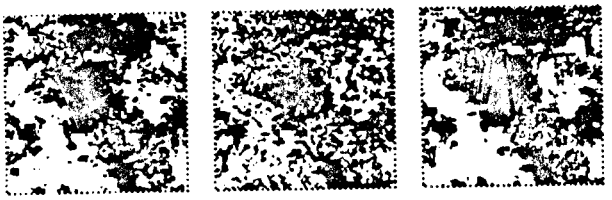
Note that this estimation algorithm allows us to reconstruct a pattern f from the noisy observations g *without having to adjust any free parameters*. The only prior assumptions correspond to the qualitative structure of the field f (first order, isotropic MRF) and to the nature of the noise process. In practice, this means that we can apply it to restore any piecewise uniform image with uniform granularity, even if it has not been generated by a Markov random process. In the particular case of a binary field sent through a BSC, we have developed a very efficient procedure for approximating the MPM estimator, which also permits us to find the optimal (Maximum Likelihood) estimate using only a one dimensional search (see Marroquin, 1985 for details). We have used this algorithm to reconstruct a variety of binary images with excellent results. In figure 4 we show such a restoration. The observations (b) were generated from the synthetic image (a) with an actual error rate of .35 (assumed unknown). The MLE for f is shown in (c).

5.2. Reconstruction of Piecewise Continuous Functions.

In this section we will illustrate the application of the local spatial interaction models and estimation techniques that we have described to the reconstruction of piecewise continuous functions from noisy observations taken at sparse locations.

In this reconstruction, it will be important not only to interpolate smooth patches over uniform regions, but to locate and preserve the discontinuities that bound these regions, since very often they are the most important parts of the function. They may represent object boundaries in vision problems (such as image segmentation; depth from stereo; shape from

Figure 3. (a) Original ternary MRF. (b) Noisy observations (additive Gaussian noise). (c) Optimal (Maximum likelihood) estimate.



(a) (b) (c)

(a) (b) (c)

Figure 4. (a) Synthetic image. (b) Noisy observations. (c) Maximum Likelihood Estimate.

shading; structure from motion, etc.); geological faults in geophysical information processing, etc.

s we mentioned in section 1.4, an approach to this problem (see Terzopoulos (1984)) consist of, first, interpolating an everywhere smooth function over the whole domain; then, applying some kind of discontinuity detector (followed by a thresholding operation) to try to find the significant boundaries, and finally, to re-interpolate smooth patches over the continuous subregions.

The results that have been obtained with this technique, however, are not completely satisfactory. The main problem is that the task of the discontinuity detector is hindered by the previous smooth interpolation operation. This becomes critical when the observations are sparsely located, since in this case, the discontinuities may be smeared in the interpolation phase to such a degree that it may become impossible to recover them in the detection phase.

In contrast, in the Bayesian approach, the boundary detection and interpolation tasks are performed *at the same time*. To apply the general recontruction algorithms developed above to this problem, the main issue is the representation of the concept of "piecewise continuity" in the form of a prior Gibbs distribution in a meaningful way.

A flexible construction involves the use of two coupled MRF models: one to represent the function (the "surface") itself, and another to model the curves where the field is discontinuous. A coupled model of this kind was first used by Geman and Geman (1984) in the context of the restoration of piecewise constant images. Terzopoulos (1985) has recently attempted to translate this idea in the continous and deterministic framework of standard regularization.

This model can be adapted to our problem by modifying the choice of the potentials and the neighborhood structure of the coupled MRF's. Specifically, the following modifications are needed:

1. Since in our case the observations are sparse, it becomes necessary to expand the size of the neighborhoods of the line field, to prevent the formation of "thick" boundaries between the smooth patches (i.e., adjacent, parallel segments of active lines in these regions). In particular, we propose that the dual lattice be 8-connected, with non-zero potentials for the cliques of the form illustrated in figure 5 (a) and (b). The inclusion of the cliques of figure 5-b has the additional advantage of penalizing the occurance of sharp turns, permitting us to model the formation of piecewise smooth boundaries using a binary line process instead of the 4-valued process proposed by Geman and Geman. The potentials for these cliques are computed in the following way:

Let V_a, V_b denote the potentials associated with the cliques C_a, C_b of figure 5 (a) and (b), respectively, and let S_k ($k \in \{a, b\}$) denote the number of line elements belonging to C_k that are "on" at a given time, i.e.,

$$S_k = \sum_{i \in C_k} l_i \quad , \quad k = a, b$$

The potentials V_k are given by:

$$V_k = \beta \phi_k(S_k) \quad , \quad k = a, b$$

Figure 5. Cliques for the line process

(a) (b)

where β is a constant, and the functions ϕ_k are defined by the following tables:

S_a	0	1	2	3	4
ϕ_a	0	0.4	0.25	1.2	2.0

S_b	0	1	2
ϕ_b	0	0	10

It is not difficult to see that this choice of potentials will effectively discourage both the formation of thick boundaries ($S_b = 2$) and the presence of sharp turns ($S_a = 3$ and/or $S_b = 2$).

2. The potentials of the depth process, which is now continuous–valued, have to be modified to express the more relaxed condition of piecewise continuity (instead of piecewise constancy). Specifically, we propose:

$$V_f(f_i, f_j, l_{ij}) = \begin{cases} (f_i - f_j)^2(1 - l_{ij}), & \text{for } |i - j| = 1 \\ 0, & \text{otherwise} \end{cases}$$

(note that $l_{ij} \in \{0, 1\}$)

3. Unlike the case of piecewise constant surfaces, we now have to worry about the maximum absolute difference in the values of two adjacent depth sites that we are willing to consider as a "smooth" gradient (and not a discontinuity). This value, which in general is problem–dependent, determines the magnitude of the constant β in equation (2), which can be interpreted as the coupling strength between the two processes.

Assuming that the observations are corrupted by i.i.d. Gaussian noise, we get the following expression for the posterior energy:

$$U_f(f, l; g) = \frac{1}{T_0} \sum_{i,j} (f_i - f_j)^2(1 - l_{ij}) +$$

$$+ \frac{1}{2\sigma^2} \sum_{i \in S} (f_i - g_i)^2 + \sum_{C_a} V_a(l) + \sum_{C_b} V_b(l)$$

where S is the set of sites where an observation is present. As a performance criterion we will use a mixed cost functional of the form:

$$e_m(f, l, \hat{f}, \hat{l}) = \sum_{i \in L_f} (f_i - \hat{f}_i)^2 + \sum_{j \in L_l} (1 - \delta(l_j - \hat{l}_j))$$

where L_f, L_l denote the depth and line lattices, respectively. This error criterion means that the reconstructed surface should be as close as possible to the true (unknown) surface, and that we should commit as few errors as possible in the assertions about the presence or absence of discontinuities.

Applying the results of section 3, we find that the optimal estimators will be the *posterior mean* for f and the *maximizer of the posterior marginals* for l.

There is one serious difficulty that prevents us from applying directly the general Monte Carlo procedure that was derived above to the computation of these optimal estimates: since the depth variables are continuous–valued, if we discretize them finely enough to guarantee sufficient precision of the results, the computational complexity of either the Metropolis or Gibbs Sampler algorithms will be very large. One way around this difficulty is to note that for any fixed configuration of the line field, the posterior energy becomes a non–negative definite quadratic form:

$$U(f \mid l, g) = \sum_{i,j:l_{ij}=0} (f_i - f_j)^2 + \alpha \sum_{j \in S} (f_j - g_j)^2 + K \quad (15)$$

where α and K are constants (note that the first sum is taken only over those pairs of sites whose connecting line element is "off", and the second one over the set S). This means that the posterior distribution of the depth field is conditionally Gaussian, so that, for any fixed l, we can find the optimal conditional estimator f_l^* as the minimizer of (15).

Let us define the set F^* as:

$$F^* = \{(f, l) \ : \ f = f_l^*\}$$

It is clear that, if $\hat{f}, \hat{l}$ are the optimal estimates for our problem, we have that:

$$(\hat{f}, \hat{l}) \in F^*$$

which suggests that we can constrain the search for the optimal estimators to this set. This can be done, in principle, by replacing the posterior energy with the function:

$$U^*(l) = U(f_l^*, l)$$

(which depends only on l), and use the standard Monte Carlo procedures to find the optimal estimator $\hat{l}$. To illustrate this idea, let us consider a physical model in the next section.

5.2.1. Hybrid parallel computers

It is well known that the steady state of an electrical network that contains only (current or voltage) sources and linear resistors will be the global minimizer of a quadratic functional that corresponds to the total power dissipated as heat (Oster et al, 1971). It is therefore possible to contruct an analog network that will find the equilibrium state of the depth field for a given, fixed configuration of the line process, i.e., that will minimize the conditional energy (8) (see Poggio and Koch,

1984; also Poggio et al., 1985). This suggests a hybrid computational scheme in which the line field (whose state is updated digitally, using, say, the Metropolis or Gibbs Sampler algorithms) acts as a set of switches on the connections between the nodes of the analog network whose voltages represent the depth process. In particular, if f_i represents the voltage at node i, the hybrid network can be represented as a 4-connected lattice of nodes (see figure 6) in which:

(i) A resistance (of unit magnitude) and a switch (controlled by the line element l_{ij}) is present in every link between pairs i, j of adjacent nodes.

(ii) If an observation g_i is present at site i, a current of magnitude equal to αg_i is injected to the corresponding node, which must also be connected to a common ground via a resistance of magnitude $1/\alpha$ (see equation 8).

A direct application of Kirchoff current law shows that at each node i of this network we will have:

$$\sum_{j \in N_i} (f_i - f_j)(1 - l_{ij}) + \alpha q_i f_i = \alpha q_i g_i$$

which corresponds to the condition

$$\text{grad } U(f \mid l) = 0$$

so that the equilibrium configuration coincides with f_i^*

This scheme can be used, in principle, to construct a special purpose hybrid computer for the fast solution of problems of this type. In a digital machine, the exact implementation of this strategy will in general be computationally very expensive, since f_i^* must be computed every time a line site is updated. It is possible, however, to develop approximations which have an excellent experimental performance, and lead to efficient implementations (Marroquin, 1985). The performance of this method is illustrated in figure 7, in which we show: (with height coded by grey level) the observations (a); the initial state of the network (with all the lines turned "off") (b); the final reconstructed surface (c), and the boundaries found by the algorithm (d), for a square at height 2.0 over a background at constant height = 1.0.

6. Signal Matching

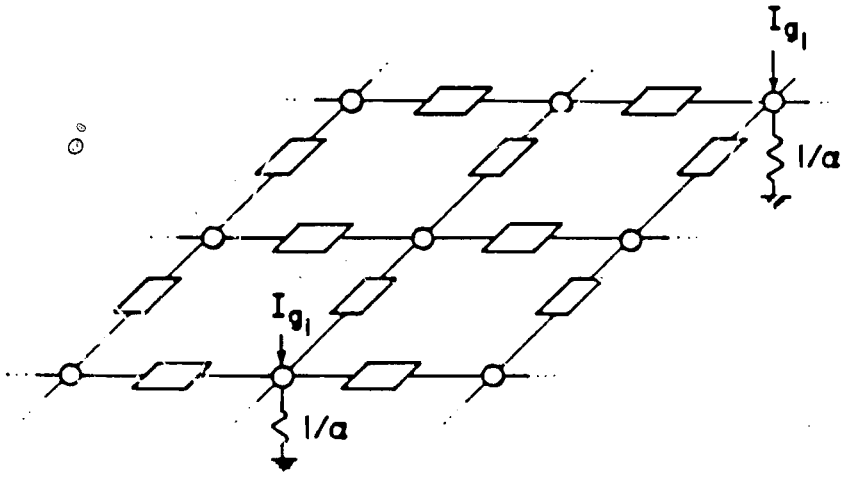
In all the estimation problems we have studied so far, the posterior energy function had the form:

$$U_P(f; g) = U_0(f) + \sum_i \Phi_i(f_i, g_i)$$

where $U_0(f)$ corresponded to the MRF model for the field f. The functions Φ_i, whose precise form depended on the particular noise model, were non-decreasing functions of the distance between f_i and g_i.

There are some cases, however, when the conditional probability distribution of the observations $P_{g|f}(g; f)$ is multimodal (as a function of f) which causes the functions Φ_i to be non-monotonic, so that the solution to the problem remains ambiguous, even if the observations are dense, and the signal to noise ratio arbitrarily high. To illustrate this situation, we will study an important instance of it: the "signal matching" problem, whose one-dimensional version is as follows:

Figure 6. Hybrid network implementing the surface reconstruction algorithm of section 4. The voltage at every node represents the height of the surface. Inside every rectangular box there is a resistance of unit magnitude and a switch whose state is controlled by the corresponding line element. (see text).



Consider two one-dimensional, real valued sequences h_L, h_R, where h_L is obtained from h_R by shifting some subintervals according to the "disparity sequence" d:

$$h_L(i) = h_R(i + d_i)$$

with

$$d_i \in Q = \{-m, -m+1, \ldots, -1, 0, 1, \ldots, m\}$$

The signal matching problem is to find d given h_L, h_R. (In a more realistic situation, we do not observe h_L, h_R directly, but rather some noise-corrupted versions g_L, g_R). Some interesting instances of this problem are the matching of stereoscopic images along epipolar lines (Marr and Poggio, 1976); the computation of the dip angle of geological structures from electrical resistivity measurements taken along a bore hole, and the matching of DNA sequences.

To make the discussion more specific, we will consider a simple example, in which the sequences h_L, h_R are binary Bernoulli sequences; we will assume that the noise corruption process can be modeled as a binary symmetric channel with known error rate, and that d is known to be a piecewise constant function. A well known instance of this problem is the matching of a row of a random dot stereogram with density ρ (Julesz (1960)), when the components of the stereo pair are corrupted by noise.

The stochastic model for the observations is then constructed by assuming that the right image is a sample function of a Bernoulli process A with parameter ρ:

$$g_R(i) = A(i)$$

The left image is assumed to be formed from the right one by shifting it by a variable amount given by the disparity function d, except at some points where an error is commited with probability ϵ. Note that some regions that appear in the right image will be occluded in the left one (see figure 8). The "occlusion indicator" o_d can be computed deterministically from d in the following way:

$$\phi_d(i) = \begin{cases} 1, & \text{if } d_{i-k} \geq d_i + k, \text{for some integer } k \in (0, m] \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

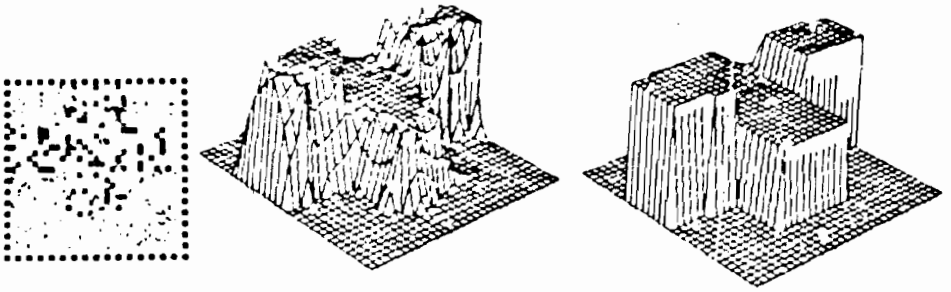
The occluded areas are assumed to be "filled in" by an independent Bernoulli process B. The final model is then:

$$g_L(i) = \begin{cases} g_R(i + d_i), & \text{with prob. } 1-\epsilon, \text{ if } \phi_d(i) = 0 \\ 1 - g_R(i + d_i), & \text{with prob. } \epsilon, \text{ if } \phi_d(i) = 0 \\ B_\rho(i), & \text{with prob. } 1, \text{ if } \phi_d(i) = 1 \end{cases} \tag{17}$$

Note that in the two-dimensional case, the index i denotes a site of a lattice, and therefore it can be represented as a two-vector (i_1, i_2) whose components denote the column and row of the site, respectively. To simplify the notation, we will adopt the following convention throughout this section: when a scalar is added to this vector index (as in $g_R(i + d_i)$ and d_{i+k}), it will be implicitly assumed that it is multiplied by the vector $(1, 0)$ (so that the above expressions should be understood as $g_R(i + (d_i, 0))$ and $d_{i+(k,0)}$, respectively). Using this convention, the observation model of equation (17) can be applied either to the one or to the two-dimensional cases.

Notice that even if the observations are noise-free ($\epsilon = 0$) the solution of the problem remains ambiguous, and it cannot be uniquely determined unless some prior knowledge about d (for example, in the form of a MRF model) is introduced. The use of a MRF

Figure 7. (a) Observations of 3 rectangles at heights 2.0, 3.0 and 2.0 over a background at height 1.0 (height coded by grey level; a white pixel means that the observation is absent at that point). (b) Equilibrium state of the network with all lines turned "off". (c) Optimal estimate.



(a) (b) (c)

model in the stereo matching case, corresponds to a quantification of the assumption of the existence of "dense solutions" (this term was introduced by Julesz (1960), and essentially corresponds to the assumption that the disparity d varies smoothly in most parts of the image; see also Marr and Poggio (1979)), and the use of the occlusion indicator corresponds to the "ordering constraint" (i.e., the requirement that if $i > j$, then $i + d_i > j + d_j$, see Baker (1981); we put $\phi_d = 1$ whenever this constraint is violated).

To formulate the estimation problem, we will consider the sequence g_L as "observations," while g_R will play the role of a set of parameters. Thus, from (17), we have (assuming, for simplicity that $\rho = \frac{1}{2}$):

$$P(g_L(i) = k \mid d, g_R) = P_{g|d}(k) =$$

$$= \begin{cases} 1-\epsilon, & \text{if } \phi_d(i) = 0 \text{ and } g_R(i+d_i) = k \\ \epsilon, & \text{if } \phi_d(i) = 0 \text{ and } g_R(i+d_i) \neq k \\ \frac{1}{2}, & \text{if } \phi_d(i) = 1 \end{cases}$$

As a prior model for the disparity field, we may use a first order MRF with generalized Ising potentials, such as the one presented in section 5.1. Other models may also be used, including the coupled depth and line fields that we discussed in the previous section. For the present, let us assume that the simpler Ising model is adequate. Note that even when the matching problem is one-dimensional (we are asuming that there is no vertical disparity between the images, so that the matching can be done on a row-by-row basis), the two-dimensional nature of the prior MRF model for the disparity introduces a coupling between matches at adjacent rows. The posterior energy is:

$$U_P(d; g) = \frac{1}{T_0} \sum_{i,j} V(d_i, d_j) + \frac{1}{2} \sum_i \phi_d(i) \ln 2 +$$

$$+ \frac{\alpha}{2} \sum_i (1 - \phi_d(i)) \delta(g_L(i) - g_R(i + d_i)) \qquad (4)$$

where

$$\alpha = \ln\left(\frac{\epsilon}{1-\epsilon}\right)$$

It is possible to apply the general Monte Carlo algorithms presented above to approximate the optimal estimate $\hat{d}$ with respect to a given performance measure (such as the mean squared error). Their use in this case, however, is complicated by the introduction of the occlusion function ϕ_d in the posterior energy: the size of the support for this function equals the total number of allowed values for the disparity (see equation (16)). If this number is large, the computation of the increment in energy, or of the conditional distributions (if the Gibbs Sampler is used) may be quite expensive. In many cases, however, the size of the regions of constant disparity is relatively large compared with the size of the occluded areas. In these cases, one can approximate the posterior energy by:

$$U_P(d) = \frac{1}{T_0} \sum_{i,j} V(d_i, d_j) + \frac{\alpha}{2} \sum_i \delta(g_L(i) - g_R(i + d_i))$$

and increase significantly the computational efficiency. It is also possible, particularly for the high signal to noise ratio case, to design deterministic, highly distributed algorithms for the efficient computation of the optimal estimator. The details of these designs can be found in Marroquin, 1985.

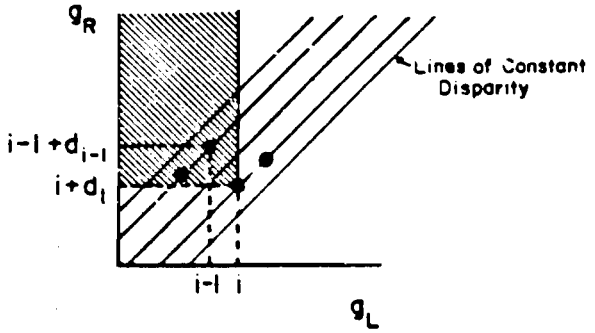
To illustrate the performance of this approach, we present in figure 9 a random dot stereogram portraying a square floating over a uniform background (panel (a)), and the reconstructed surface (panel (b)).

7. Parallel Implementations.

7.1. Connection machine architectures.

The general Monte Carlo procedure that we have presented for the approximation of the optimal Bayesian

Figure 8. Occluded Regions: The horizontal and vertical axis represent points in one row of the left and right images, respectively. Matching points are represented by black circles. Any match in the shaded region will occlude the point i



estimators of MRF's can be greatly accelerated if it is implemented in a parallel architecture. A necessary condition for the convergence of the probability measures of the Markov chains defined by the Metropolis or Gibbs Sampler algorithms to the posterior Gibbs distribution (and therefore, for the convergence of the approximations given by equations (13) and (14) to the desired estimates) is that if two sites belong to the same clique, they are never updated at the same time. It is important to note, however, that this condition is also sufficient only for the case of the Gibbs sampler: if one updates simultaneously the states of all non-neighboring sites, the reversibility of the resulting chain will be destroyed, so that it will no longer be possible to guarantee the convergence of the Metropolis algorithm to the desired result (see Marroquin, 1985).

If one implements the Gibbs sampler in a parallel architecture in which a processor is assigned to each site, the total execution time will be reduced by a factor of

$$\frac{N}{K}$$

where K is the so called "chromatic number" of the graph that describes the neighborhood structure, and it is equal to the minimum number of colors needed to color the sites of the lattice in such a way that no two neighbors are the same.

An example of such a massively parallel architecture is the "Connection Machine" (Hillis, 1985), currently under construction at Thinking Machines Corp. and at the Artificial Intelligence laboratory at MIT. This machine was originally designed for the parallel processing of structured symbolic expressions, such as frames and semantic networks. It is a "Single Instruction Multiple Data" (SIMD) array processor consisting of 256,000 processing units (each with a single bit Arithmetic/Logical unit, and about 4K bits of storage) organized in a four-connected lattice that is 512 elements square. Besides this nearest-neighbor connectivity, it will also be possible (although computationally more expensive), to connect any two processors in the array using a "Cross Omega" router network (Knight, in Winston, 1984).

At each cycle of the machine, for which we will assume a duration of one microsecond, an instruction is executed by each processor, and a single bit is transmitted to its neighbors. This means that the updating scheme can be implemented most efficiently if the field is first order Markov, but higher order processes can also be implemented without using the router by successively propagating the transmited state (the execution time, therefore, will grow linearly with the order of the field).

To make this discussion more concrete, consider, as an example, the problem of finding the optimal estimate for an M-ary, first order MRF with Ising potentials (i.e., the segmentation of a piecewise constant image) from noisy observations. Let us assume that the estimator is to be implemented in the "Connection Machine", and suppose that by the use of appropriate scaling factors, all the numbers can be represented as 16-bit integers. We will use the following conservative assumptions: We assume that 16 cycles of a single 1-bit processor are needed to perform 16-bit addition, substraction or comparison; 16^2 cycles to perform multiplication or division; 2×16^2 cycles for generating a pseudo-random number with uniform distribution on a given interval; 16 cycles for memory transfer operations, and 6×16^2 cycles for computing an exponential.

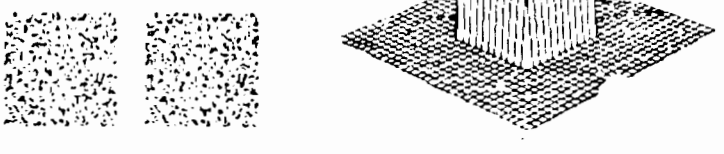
Assuming that we run 250 iterations of the system, and ignoring the overhead time we get that

$$\text{Exec. Time} \approx 1.4(M-1) \text{ seconds}$$

For the particular case of binary images, we have developed a deterministic scheme for which this execution time can be reduced by an order of magnitude (see Marroquin, 1985).

In the case of the reconstruction of piecewise smooth functions from sparse data, the optimal estimator can also be implemented in this machine. To study this implementation, we first note that the chromatic numbers of the graphs associated with the line and depth neighborhood systems are 4 and 2, respectively,

Figure 9. (a) Random dot stereogram. (b) Reconstructed surface.



(a) (b)

which means that the coupled process has a chromatic number of 6. In figure 10 (a) we illustrate one possible "coloring".

The colors of the line process are represented by the numbers 1,2,3,4, and those of the depth process by white and black circles. The updating process can be implemented in a 4–connected architecture such as the "Connection Machine", by assigning one processor to each depth site and its four adjacent line elements. We will thus have two different populations of processors, whose configurations are shown in figures 9 (b) and (c), respectively.

Each complete iteration consist on 6 major cycles: In the first two, the state of the white and black depth variables is respectively updated, and in the next four, the new states of the binary line variables stored in (say) the white processors are successively computed and transmitted to the corresponding memory locations of the neighboring black processors. Note that in this scheme we have some redundancy in the use of memory (each binary variable is stored twice), but the state of all the elements needed for each updating operation is always available from adjacent processors. Considering that the Monte Carlo algorithm requires about 200 iterations to converge, we estimate in this case an execution time of approximately 2.5 seconds, independently of the lattice size. As before, we have also developed in this case a deterministic scheme with very good experimental performance, for which the execution time can be reduced by at least an order of magnitude.

7.2. Hybrid analog-digital computers and Hopfield networks

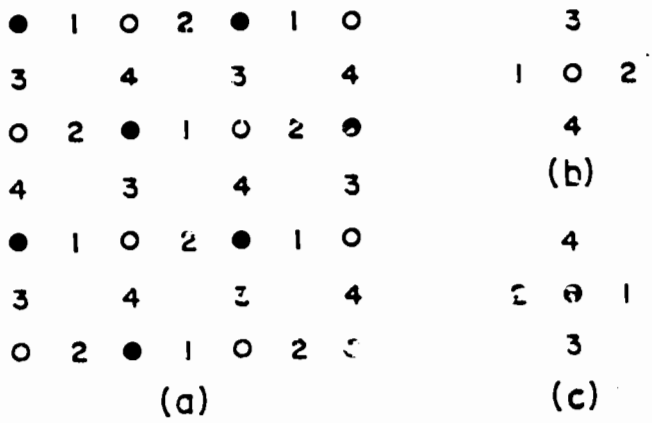
As we mentioned in section 4.2.1, the reconstruction of piecewise continuous functions can be achieved by coupling two MRFs, one corresponding to the continuous field and the other to the discontinuities. From this scheme we have suggested a special purpose parallel computer consisting of an analog network of resistances – corresponding to the continuous intensity field – and a digital network – corresponding to the line process, coupled *via* D-A and A-D converters. The idea suggested by computer experiments (Marroquin, 1985) is that the two processes can run on different time scales, a slow one for the digital part and a fast one for the analog network. In this way the two processes are effectively decoupled and the continous field finds its equilibrium effectively instantaneously after each update of the line process. Koch, Marroquin and Yuille et al. (1985) discuss implementations of this idea. This idea can be extended to *multilayered hybrid networks*, each layer corresponding to a MRF and being digital or analog depending on the continous or binary nature of the field. Hybrid multilayered architectures of this type are especially attractive for implementing the fusion of several vision processes.

Finally, we mention that Koch et al. (1985) have been experimenting successfully with a special type of analog networks – Hopfield networks – whose equilibrium states correspond to approximations of the optimal estimators.

8. Conclusions.

In this paper we have presented a probabilistic approach to the solution of a class of perceptual problems. We showed that these problems can be reduced to the recontruction of a function on a finite lattice from a set of degraded observations, and derived the Bayesian estimators that provide an optimal solution.

Figure 10. (a) Coloring of the coupled line–depth lattice. (b) and (c) Elements whose state is stored in each of the two types of processors of a 4–connected parallel architecture.



We have also developed efficient distributed algorithms for the computation of these estimates, and discussed their implementation in different kinds of hardware. To demonstrate the generality and practical value of this approach, we studied in detail several applications: the segmentation of noise-corrupted images; the recontruction of piecewise smooth surfaces from sparse data and the reconstruction of depth from stereoscopic measurements.

8.1. Connection with Standard regularization

The maximum *a posteriori* (MAP) estimate of a MRF is obviously similar to a variational principle of the general form of equation (3), since the use of this criterion defines the optimal estimator as the **global** minimizer of the posterior energy U_P (equation 6): the first term measures the discrepancy between the data and the solution, the second term is now an arbitrary "potential" function of the solution (defined on a discrete lattice). It is then natural to ask for the connection between standard regularization principles and the MRF approach. It turns out that a MAP esimate leads to the minimization of a functional U_P – in general not quadratic – that reduces to a quadratic functional, of the standard regularization type, when the MRF is continuous-valued, the noise is additive and gaussian (the term $\sum \Phi_i(f, \eta_i)$ will be quadratic) and first order differences of the field are zero-mean, independent, gaussian random variables (thus the a priori probability distribution is a Gibbs distribution with quadratic potentials so that the term the term $U_0(f)$ is quadratic).

8.2. The Fusion problem

This approach also permits, in principle, the incorporation of more than one modality of observations into a single estimation process, as well as the simultaneous estimation of several related functions from the same data set. This makes one hope that this framework could be useful in the solution of difficult problems that require such an integrated approach.

For instance, the stereo matching problem in real situations has not been solved yet in a completely satisfactory way. The same can be said of other related perceptual problems such as edge detection; image segmentation; the recovery of the shape of an object from a single two-dimensional image (the "shape form shading" problem), and the segmentation of a scene into distinct objects, as well as the recovery of their three-dimensional structure from the analysis of images formed at successive instants of time (the "structure from motion" problem). All these problems are obviously related, and it is intuitively clear that the individual solutions that can be obtained should improve if the mutual constraints that the solution of each individual problem imposes on the others were taken into account. Thus, the presence of a brightness edge should increase the likelihood of a depth edge, and viceversa; the depth estimated from stereo should be compatible with the shape derived from shading; points belonging to the same region in an image should move together, etc. We believe that these constraints can be incorporated in the potential functions of the corresponding MRF models, so that the combined optimal estimation process represents, in fact, an integrated cooperative solution to these problems, with, hopefully, a significantly improved performance.

READING LIST

Abend, K. "Compound decision procedures for unknown distributions and for dependent states of nature," in **Pattern Recognition**, L. Kanal, ed., Thompson Book Co. Washington, D.C. (1968).

Barrow, H.G. and Tennenbaum, J.M. "Interpreting line drawings as three dimensional surfaces," *Artificial Intelligence*, **17**, 1981.

Brady, J.M. *Computing Surveys*, **14**, (1982).

Brown, C.M. *Science*, **224**, (1984)

Besag, J. "Spatial interaction and the statistical analysis of lattice systems,"J. Royal Stat. Soc. B 34 75-83 (1972).

Cohen, F.S., and D. B. Cooper. "Simple parallel hierarchical and relaxation algorithms for segmenting noncausal Markovian random fields," Brown University Laboratory for Engineering Man/Machine Systems, Tech. Report LEMS-7 (1984).

Cross, G.C. and A. K. Jain. "Markov random field texture models," IEEE *Trans. Pattern Analysis and Machine Intelligence*, **5** (1983).

Elliot, H., R. Derin, R. Christi, and D. Geman, "Application of the Gibbs distribution to image segmentation," Univ. of Massachusetts Technical Report (1983).

Feller, W. **An introduction to probability theory and its applications**, Vol I. John Wiley and Sons, New York (1950).

Gallager, R. G. **Information theory and reliable communication** John Wiley and Sons, New York (1968).

Geman, S. and D. Geman. "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images," IEEE *Trans. Pattern Analysis and Machine Intelligence*, **6**, (1984).

Grenander, U. "Tutorial in Pattern Theory," Div. of Applied Math. Brown University (1984).

Grimson, W.E.L. **From Images to Surfaces** MIT Press, Cambridge, Mass. (1981).

Grimson, W.E.L. "A computational theory of visual surface interpolation," *Phil. Trans. R. Soc. London B*, 298, (1982a).

Habibi, A. "Two dimensional Bayesian estimation of images," *Proc. IEEE* 60, (1972).

Hansen, A.R. and H. Elliot. "Image segmentation using simple Markov field models," *Comp. Vision, Graphics, and Image Proc.* 20, (1982).

Hassner, M. and J. Sklansky. "The use of Markov random fields as models of texture," *Comp. Vision, Graphics and Image Proc.* 12, (1980).

Hillis, D. "The Connection Machine," M.I.T. Department of Electrical Engineering and Computer Science Ph.D. Thesis (1985).

Julesz, B. "Binocular depth perception of computer generated patterns," *Bell Sys. Tech. J.* 39 (1960).

Kashyap, R.L., and R. Chellappa. "Estimation and choice of neighbors in spatial interaction models of images," *IEEE Trans. on Info. Theory* 29 (1983).

Kemeny, J.G., and J. L. Snell. **Finite Markov Chains** Van Nostrand, New York (1960).

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing,"*Science* 220 (1983).

Koch, C., J. Marroquin, and A. Yuille. "Analog 'neuronal' networks in early vision," Massachusetts Institute of Technology Artificial Intelligence Laboratroy Memo 751 (1985).

Marr, D. **Vision,** *A computational investigation into the human representation and processing of visual information*, W. H. Freeman & Co., San Francisco, 1982.

Marr, D. and T. Poggio. "From understanding computation to understanding neural circuitry," *Neur. Res. Bull.* 15 (1977).

Marroquin, J. "Surface reconstruction preserving discontinuities," Massachusetts Institute of Technology Artificial Intelligence Laboratory Memo 792 (1984).

Marroquin, J. "Probabilistic solution of inverse problems," Ph.D. Thesis, Massachusetts Institute of Technology (1985).

Metropolis, N. et al. "Equation of state calculations by fast computing machines,"*J. Phys. Chem.* 21 (1953).

Morozov, V.A. **Methods for Solving Incorrectly Posed Problems**, Springer-Verlag, New York, 1984.

Nahi, N.E. and T. Assefi. "Bayesian recursive image estimation," *IEEE Trans. on Computers* 21 (1972).

Oster, G.F., A. Perelson and A. Katchalsky. "Network Thermodynamics," *Nature* 234 (1971).

Poggio, T. "Vision by man and machine," M. I. T. Artificial Intelligence Laboratory Memo 776 (1984).

Poggio, T. and C. Koch. "Analog networks: a new approach to neural computation," M. I. T. Artificial Intelligence Laboratory Memo 783. (1984).

Poggio, T. and V. Torre. "Ill-posed problems and regularization analysis in early vision," M. I. T. Artificial Intelligence Laboratory Memo 773 (1984).

Poggio, T., V. Torre, and C. Koch. "Computational vision and regularization theory," *Nature* 317 (1985).

Poggio, T., H. Voorhees, and A. Yuille. "Regularizing edge detection," M. I. T. Artificial Intelligence Laboratory Memo 776 (1984).

Terzoupulos, D. "Multiresolution computation of visible-surface representations." Ph. D. Thesis M.I.T. Department of Electrical Engineering and Computer Science (1984).

Terzoupulos, D. "Integrating visual information for multiple sources for the cooperative computation of surface shape," to appear in **From Pixels to Predicates: Recent Advances in Computational and Robotic Vision**, ed. A. Pentland, Ablex, (1985).

Tikhonov, A.N., and V. Y. Arsenin. **Solutions of Ill-Posed Problems**, Winston and Sons, New York (1977).

Winston, P. "Proposal to DARPA," M.I.T. (1984).

Wong, E. "Two-dimensional random fields and the representation of images," *SIAM J. App. Math.* 16, 4 (1968).

Woods, J. W. "Two-dimensional discrete Markovian fields,"*IEEE Trans. Info. Theory* 18 (1972).

GC976145 35

Stereo Verification in Aerial Image Analysis

David M. McKeown, Clifford A. McVay,
and Bruce D. Lucas

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pa. 15213

Abstract

This paper describes a flexible stereo verification system, STEREOSYS, and its application to the analysis of high resolution aerial photography. Stereo verification refers to the verification of hypotheses about a scene by stereo analysis of the scene. Unlike stereo interpretation, stereo verification requires only coarse indications of three-dimensional structure. In the case of aerial photography, this means coarse indications of the heights of objects above their surroundings. This requirement, together with requirements for robustness and for dense height measurements, shape the decision about the stereo system to use. This paper discusses these design issues and details the results of an implementation.

Subject Terms: Computer Vision, Stereo Analysis, Stereo Verification, Aerial Photo Interpretation, Artificial Intelligence, Knowledge Based Interpretation, Image/Map Databases

1. Introduction

This paper describes a flexible stereo verification system, STEREOSYS, and its application to the analysis of high resolution aerial photography. Stereo verification refers to the verification of hypotheses about a scene by stereo analysis of the scene. Unlike stereo interpretation, stereo verification requires only coarse indications of three-dimensional structure. In the case of aerial photography, this means coarse indications of the heights of objects above their surroundings. This requirement, together with requirements for robustness and for dense height measurements, have shaped the decision about the stereo system to use.

In this research we have attempted to address stereo analysis in a very unconstrained environment. Rather than simply focusing on isolated image analysis where stereo pairs are carefully controlled, we have constructed a system that can automatically perform matching and analysis using arbitrarily selected images. We are motivated by the observation that if knowledge-based image understanding systems are to begin to perform analysis tasks at a level of performance required for mapping and photo interpretation, they must be able to accommodate a much broader range of task uncertainty and complexity than has been previously demonstrated in any research or development system.

Stereo verification deals with a variety of problems that are not ordinarily present in isolated experiments with stereo matching and analysis. Some of the most interesting problems involve:

- The selection of an appropriate conjugate image pair from a database of overlapping images based on criteria that would maximize the likelihood for good correspondence.
- The image pairs must be dynamically resampled such that the epipolar assumption (ie., epipolars are scan lines) used in most region-based stereo matching algorithms can be applied.
- Because the size of the areas to be matched varies greatly, the system design must be flexible and general.
- An initial coarse registration step is generally necessary because the quality of the correspondence between conjugate pairs varies greatly. In many cases the magnitude of the initial misregistration is greater than the expected disparity shift.
- The system must analyze the stereo results and generate a symbolic description that provides an estimate of the actual height of the region in question, and the confidence of that estimate. The computation of a depth map (disparity map) is not a sufficient final result.

These requirements, in turn, raise a set of broader research issues:

1. How can an aerial image database be used to automatically generate a useful stereo pair containing an arbitrary region?
2. How can a stereo system handle the misregistration problems inherent in variable sourced image databases?
3. What kind of stereo results are appropriate for use in a verification process?
4. How can stereo results be analyzed so as to reflect not only the presence (or absence) of height but also the inherent reliability of the results?
5. How useful is stereo verification within a knowledge-based image interpretation system? What constraints does it provide, and how important is stereo information in producing accurate scene descriptions?

The results of this research indicate that image/map database issues in stereo verification influence the utility of such an approach as much as the underlying stereo matching algorithm. In fact, they are intimately related. The ability to be flexible in the selection of stereo pairs provides opportunities for multi-temporal, multi-scale, or multi-look matching. Equally as important is flexibility in the matching algorithm, especially with respect to assumptions that require nearly perfectly aligned conjugate images, a situation that is unlikely to occur in outside of the laboratory.

We believe that the ability to dynamically select conjugate image pairs from a database based upon the region of interest and knowledge of the requirements of the matching algorithm is required for a fully automated image analysis system. Our results also indicate that stereo analysis can function as a very powerful discriminator in an image understanding system without having to perform 3D shape reconstruction. That is, coarse estimates of height, coupled with confidence in those estimates, can greatly constrain search during image interpretation.

This paper discusses these broader research issues as well as providing the reader with an analysis of the results of our experimentation and details of the actual implementation.

2. Stereo Verification in SPAM

STEREOSYS was developed as a knowledge source for SPAM[1,2], a rule-based system that uses knowledge from a variety of sources to interpret airport scenes in aerial imagery. Many of the requirements for flexibility in a stereo system arise directly from the fact that STEREOSYS must interact in a larger context, that of the image understanding system. As we move from isolated computer vision experiments to system integration, the performance of particular components must be evaluated within the constraints and context of the overall system. SPAM manages and invokes various specialized low-level image analysis processes that allow it to gather information about regions in the image. These processes include texture analysis, feature alignment and grouping[3], and depth cue generation. SPAM has developed along two lines:

- The addition and refinement of knowledge about airports and procedures for recognition and matching of image-based descriptions to the airport scene model.
- The addition and refinement of low-level image processes that support the SPAM control structures by providing primitive intermediate-level scene descriptions.

STEREOSYS falls into the latter category as it uses stereo to generate a depth map (disparity image) description given a hypothesis region in the image. The role of STEREOSYS in the overall system is to verify hypotheses such as *terminal building, access road, tarmac, parking apron,* and *hangar* by measuring the amount of disparity within a hypothesis region and thereby estimating the likelihood that the region is above or at the ground plane. Further, if the region is deemed to be mostly above the ground, STEREOSYS provides a coarse estimate of the absolute height above the ground. One may contrast this with methods for stereo reconstruction that use feature matching or segment-based techniques: STEREOSYS does not attempt to construct a precise three-dimensional model of the feature within the scene. For the tasks that SPAM requires, for example, the verification of a hangar hypothesis, it is not as important to determine the shape of the roof as much as to reliably determine whether a roof of some type is present. The issue of robustness and reliability in aerial image interpretation is of principal importance since most of the hypotheses generated by the system will not correspond to features in the scene having significant height. Therefore, the ability to refute incorrect hypotheses such as hangar and terminal building by determining there is no apparent height as well as to reliably confirm 'no height' hypotheses in areas such as tarmac and parking aprons puts performance expectations on the stereo system that transcend simple stereo matching.

SPAM invokes STEREOSYS as a result of recognizing one of two situations. First, as a part of low-level information gathering, we might want to test every region generated by the segmentation system having certain shape and size properties to determine whether it has significant height above the ground plane. Second, as a part of high-level disambiguation, there are a variety of cases where spatial constraints derived from the rule-based airport model are unable to distinguish between two competing hypotheses. For example, assume SPAM has found a conflict between two interpretations, "terminal building" and "parking lot". Spatial knowledge would allow these hypotheses to occupy similar spots in the overall scene for a wide variety of airports and, therefore, would not be able alone to resolve the conflict. Another common example are compact two-dimensional regions, such as runup pads and the roofs of maintenance buildings. Shape and size metrics such as compactness and area provide only weak cues in this situation. SPAM specifically recognizes situations where competing hypotheses involve features that can be disambiguated based upon knowledge of their height relative to their surroundings. Since we may often be looking at regions that are primarily at the ground plane, the ability to reliably determine that there is no apparent height difference between the region and its neighborhood is equally important.

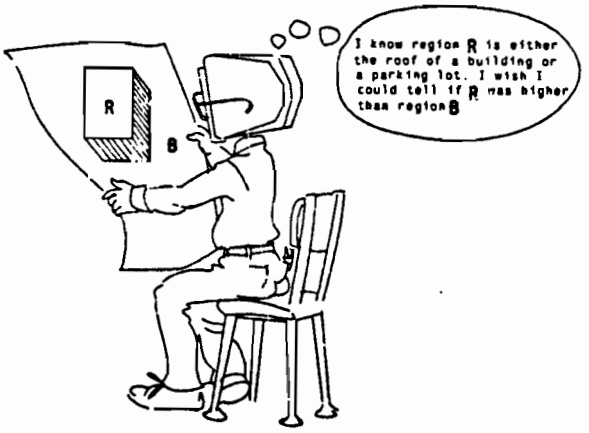


Figure 2-1: Stereo Verification

In either case, the stereo verification process can be characterized as follows:

1. Given a region *R1* within a geographic area *A1* from image *I1*, find an appropriate second image *I2* that contains a geographic area *A2* that is the same as geographic area *A1*. STEREOSYS has access to a database of images through primitives provided by the MAPS system[5,6].
2. Image fragments *A1* and *A2* are rectified (warped) and registered (shifted/rotated) into a stereo pair of overlaying geographic rectangles *W1* and *W2*.
3. The *W1*-*W2* stereo pair is processed and the result is analyzed in order to compute confidence values that measure the height of *R1* relative to its surroundings along with the system's overall confidence in the stereo result.

In the remainder of this paper we will discuss the stereo matching algorithm, how STEREOSYS uses this algorithm to perform stereo verification, and some experimental results that illustrate the strength of this technique as well as some of the more interesting pragmatic problems encountered in complex aerial imagery. Section 3 describes the basic stereo matching process used by STEREOSYS. Section 4 describes the interaction and communication between the image analysis system, SPAM, the image/map database system, MAPS, and the stereo verification system, STEREOSYS. Section 4 also gives the sequence of steps necessary to apply the stereo algorithm to an arbitrarily selected region of an image. Section 5 shows examples of preliminary experiments with SI; the effects of good and poor initial correspondence estimates, the effect of the fine registration step on the subsequent matching, and the evaluation of STEREOSYS over many test regions. Section 6 overviews the strengths and limitations of this work, and suggests future research directions.

3. The Stereo Process

STEREOSYS uses a stereo matching program, SI, described in detail elsewhere[7]. In this Section we will review this stereo matching algorithm. SI produces a *disparity image (map)* that is registered to the Left stereo pair image and whose pixel values indicate the film plane displacement of matched points in the stereo pair. The disparity value is in one-to-one correspondence with distance, or depth, from the camera and therefore indicates relative height in vertical aerial

photography. The process, in effect, correlates neighborhoods about every pixel, but uses the method of differences to avoid costly exhaustive searches.

3.1. Method of Differences

Let $I_1(x,y)$ and $I_2(x,y)$ denote the two images of a stereo pair, and let $h(x,y)$ denote the disparity map. Then the values of the disparity map are a statement that the point (x,y) in I_1 matches the point $(x+h(x,y),y)$ in I_2, that is that

$$I_1(x,y) = I_2(x+h(x,y),y)$$

Let $\hat{h}(x,y)$ denote the correct disparity map. The process begins with a uniform disparity map $h_0(x,y)$, and successively updates the disparity map, yielding h_1, h_2, etc. Ideally, as successive refinements proceed, $h_k \rightarrow \hat{h}$.

Consider a point (x,y) in the left image of the stereo pair; the difference $\hat{h}(x,y) - h_0(x,y)$ between the correct disparity value and our initial estimate is the amount by which the stereo process must correct the disparity in going from h_0 to h_1. Initially this difference will be relatively large because the uniform disparity estimate is not particularly accurate. Because of this, the method of differences requires that we start out with smoothed images to accommodate these large differences. As the disparity estimate h_k improves, we can use less smoothed images because the error between h_k and $\hat{h}$ decreases.

Suppose we have computed a disparity map h_k; that is, we estimate that the point (x,y) in I_1 matches the point $(x+h_k(x,y),y)$ in I_2. To compute h_{k+1}, we wish to adjust the disparity at each point (x,y) by an amount $\delta(x,y)$ so that the difference between the images is made as small as possible, that is

$$I_1(x,y) - (I_2(x+h_k(x,y)+\delta(x,y),y))$$

is minimized. Minimizing this quantity directly involves a costly search over the possible values of δ. Instead, the method of differences estimates this quantity by using derivatives:

$$I_1(x,y) - (I_2(x+h_k(x,y),y) + \delta(x,y) D_x I_2(x+h_k(x,y),y))$$

where D_x denotes derivative w.r.t. x.

This quantity is linear in $\delta(x,y)$, as illustrated in Figure 3-1. It could be minimized directly, but we get better results by combining many such estimates from each point in the neighborhood of (x,y) using a least squares technique, and then minimizing. In any case, the estimate based on derivatives is valid only over a range around $x+h$ on the order of the size of the averaging window that has been used to smooth the image. But to be useful we require that this estimate be accurate over a range of at least δ, the discrepancy between the actual disparity and our disparity estimate. Thus because the initial disparity error is large, we must start with relatively smoothed images. For example, some of our images require an adjustment on the order of 15 pixels between the initial disparity estimate and the actual disparity, and so STEREOSYS begins with 32 by 32 smoothing windows.

3.2. Some Pragmatic Issues in Stereo Matching

SI is also capable of computing a global registration shift between a stereo image pair, also by the method of differences. That is, a global offset can be obtained that indicates how much one image is translated, or shifted, relative to the other. This capability can often salvage the analysis of misregistered stereo pairs and is very attractive for use with SPAM since the underlying MAPS database does not have the image control necessary to guarantee accurately registered stereo pairs.

SI does not involve the use of sensitive feature extraction thresholds. Stereo matching in SI is accomplished for every pixel and is not restricted to selected image features such as interesting areas[8], edges[9,10] or other extracted features[11]. Limiting a stereo procedure to matching extracted image features makes the process sensitive to the extraction technique and its associated thresholds. Since SPAM will be using a stereo process over a wide range of images and regions, such extraction thresholds should be avoided wherever possible.

Another issue in the selection of SI for use by SPAM has to do with the fact that SPAM is not using stereo to recognize objects or build conceptual models from the stereo results. SPAM simply wants to know if the region of interest has height relative to its surroundings. A dense disparity image registered to the image containing the region of interest is an ideal source of data for the analysis necessary to do simple height verification. Almost all other stereo processes we are aware of produce sparse disparity results designed for purposes other than verification. Work by Panton[12] and Henderson[13] provide possible exceptions.

In summary, unlike many other stereo processes, SI is not overly reliant on perfectly registered stereo pairs taken simultaneously by well parameterized cameras, nor does it require threshold tweaking to accommodate matching of edges or vertices. It produces an easily analyzed dense disparity image. SI was chosen for use in stereo verification because these properties coincide well with the aerial image analysis domain that SPAM addresses.

4. Using Stereo Verification with an Aerial Image Database

Certain steps are necessary for a stereo process to work automatically as a verification procedure in association with a database of aerial imagery. A block diagram is given in Figure 4-1 that outlines the procedure and shows the interactions between the Image Analysis Process (SPAM) and the Image Database (MAPS). We can loosely organize these steps, beginning with the identification of a region of interest by the image analysis process as:

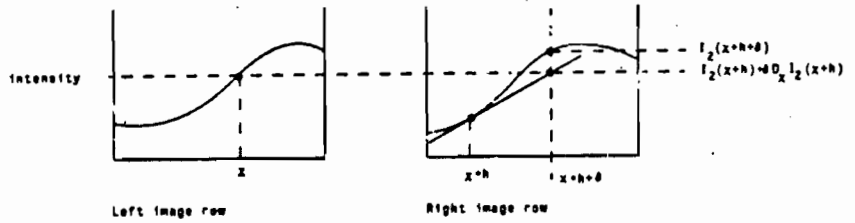


Figure 3-1: Estimating Disparity

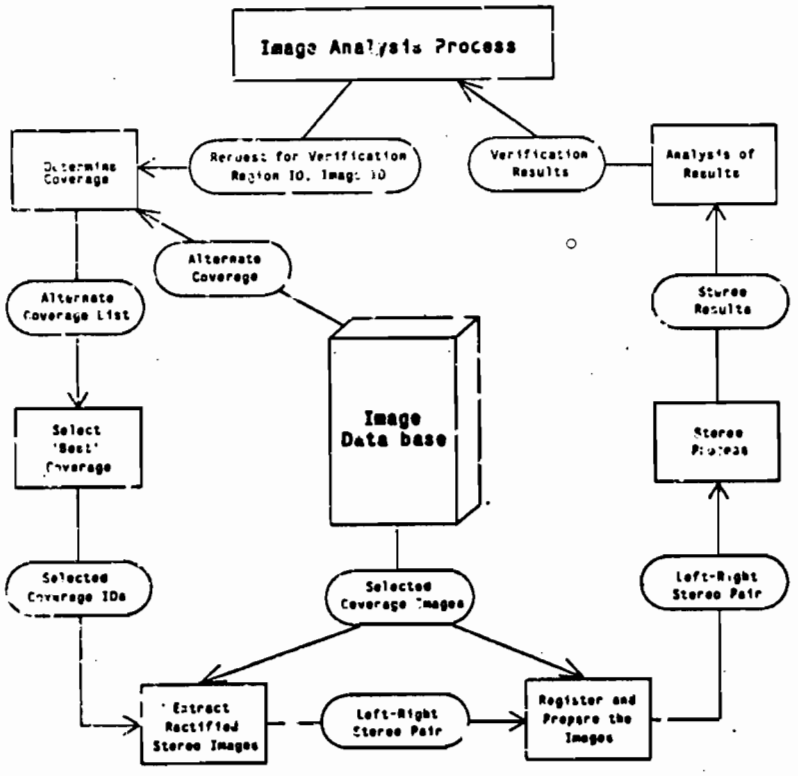


Figure 4-1: The Stereo Verification Process

1. Select Coverage: Determine the available alternate images that cover the region of interest. Select the most appropriate alternative(s).
2. Extract the Stereo Pair: Extract a stereo pair of the region from the image coverage selected.
3. Register the Stereo Pair: Compensate for misalignment errors inherent in the aerial image database. Do any other processing necessary to assure the stereo pair meets any assumptions made by the stereo process.
4. Run the Stereo Process. Apply some stereo matching process (eg., SI).
5. Analyze the Results: Analyze the stereo results in order to verify if the region of interest has height relative to its surroundings.

The STEREOSYS process is initiated by SPAM with parameters identifying the region of interest, the database image that is being interpreted and contains the region, and an estimated height range (0-5 meters, 0-15 meters, 10-20 meters, etc) for the region.

Using the identity of the region of interest, STEREOSYS extracts the region's centroid, its boundary point list and an associated minimum bounding rectangle (MBR) from the MAPS database. This data is used in determining alternate imagery coverage, in extracting the stereo pair, and in analyzing the stereo results.

The MAPS database is used to produce an unsorted list of images, called a *coverage file*. Each image in the coverage file contains the region of interest. The image being interpretated by SPAM, and an image from the coverage file form the stereo pair.

The estimated height is used to select a *disparity range* that affects the contrast of the disparity image produced by the SI algorithm. The resulting disparity image is quantized to 256 disparity levels. If this range is set too large, the disparity image will lack contrast and will be more difficult to analyze. If it is set too small, extremely large height disparities will occur outside the image range and will effectively be invisible. In other words, the initial disparity range determines the scaling of measured disparity into the disparity image. As in any linear scaling operation, one would like to utilize the full dynamic range of the output image while avoiding clipping at either end of the range. The selection of the disparity range constitutes the only external parameterization necessary in the implemented process. Our experience has shown that the disparity range need only be within a set of rather broad values to obtain useful results. For now, we use only three pre-selected ranges. Since SPAM actually selects the disparity range based on its region hypothesis, there is potential to add ranges to accommodate additional hypothesis types or to run the stereo process over a set of disparity ranges.

The following Sections will discuss these procedural steps in more detail and describe how STEREOSYS implements them. Some details are specific to the SI matching algorithm used by STEREOSYS but are mentioned so that the reader may better understand our results.

4.1. Select Coverage

The MAPS database is used to produce an unsorted list of images, called a coverage file. Each image in the coverage file contains the region of interest. The interpretation image is used to create the Left stereo pair image since the SI disparity image result overlays the Left image and, as will be seen, since there is no guarantee that the stereo image extracted from the alternate image will be properly registered

the region. The coverage file is used to select the database image from which the Right stereo pair image will be extracted. However, in most cases, the coverage file lists several images that contain the region in question. Several considerations enter into the choice of the best candidate. First, to minimize resampling extrapolation the candidate should be of the same or larger scale. Second, to reduce possible perspective distortion, the candidate should have the region of interest as near to its center as possible. In the case of vertical aerial photography this is the region's *nadir distance*. Third, if possible, the candidate should be from the same photographing mission, even flight line, as the original image to reduce temporal changes such as lighting, cloud cover, and ground movement. Figure 4-2 illustrates a pair of typical mapping aircraft flightlines that generate stereo coverage on successive frames of the same flightline as well as between adjacent flightlines. Figure 4-2 also illustrates that small changes in the aircraft platform position and direction can effect the actual area of overlap and must be accommodated; one cannot assume a constant direction or viewing position. This is discussed in Section 4.2.

Other issues such as the source of the image, its recency, the processing and digitization history can enter into the selection of the images used to produce the stereo pair. For our purposes, STEREOSYS sorts the coverage file into a best stereo coverage order with respect to the hypothesis region's originating image as follows:

- Same Mission images (sorted by nadir distance)
- Same Scale images (sorted by nadir distance)
- All Other images (sorted by nadir distance)

The first image in the sorted coverage file best satisfies these criteria and is used to create the Right image.

4.2. Extract the Stereo Pair

The extraction of the stereo pair images is not a simple subimage cropping procedure. Like almost all stereo algorithms, SI assumes image scanlines in the stereo pair are stereo epipolar lines. Without rotation this will not be the case with the selected Left and Right images. Photographic mission flight lines need not align with image digitization scanlines and, even if they did, sometimes the best coverage is found across mission flight lines or even from separate missions. For these reasons, a baseline orientation between the stereo pair is calculated so that the pair can be rotated to properly align the scanlines to meet the epipolar constraint.

However, this necessary rotation doesn't correct for distortions due to non-parallel camera axes. Even if the stereo process is sophisticated enough to account for large amounts of perspective distortions, chances are it will not be able to account for these distortions after they have been rotated. Therefore, the stereo pair Left-Right images are extracted through an orthographic rectification process before they are rotated. This method of subimage extraction removes perspective distortions by warping the subimage into a rectangular geographic box as well as establishing a common orientation for the image scanlines.

Several issues are considered in determining the size of the image area to be extracted. First, the area must contain the region's MBR plus a portion of the surrounding area since the SI stereo results will only contain relative height information. In addition, the extracted area must be large enough so that the region of interest is contained in a rectangular sub-image cropped from the rotated image.

Specifically, to produce the necessary stereo pair, STEREOSYS extracts orthographically rectified areas identified as North-South oriented geographic rectangles by sub-pixel interpolation[14]. The corners of the extraction rectangle are calculated as a function of the region's centroid, the region's MBR, the Left-Right image scales, and the rotation necessary to make the extracted image East-West scanlines align with the baseline between the database coverage images.

4.3. Register the Stereo Pair

As mentioned in Section 3, SI is capable of determining a global disparity or offset between stereo pairs. Using this SI capability, the initially extracted stereo sub-image pairs are repeatedly processed by SI to determine the local horizontal and vertical offset between the Left and Right images. With each pass over the image pair, SI calculates a global offset value between the images. The process is repeated and the offset compounds until the offset stabilizes or begins to oscillate. Calculation of the registration offset is necessary because geodetic position correspondence control between images stored in the MAPS database is not sufficiently accurate to guarantee that the extracted Right image will overlay the Left image within the tolerances over which SI can perform effective matching. As mentioned earlier, in many cases the initial registration errors may range from 5 to 30 pixels while the disparity shift is generally smaller than 10 pixels.

One can view the stereo matching process as first applying a coarse registration, followed by the actual calculation of disparity. It is interesting to note that the same technique, method of differences, appears to be effective for both global registration and local matching. A possible alternative to this registration step would be the addition of sufficient ground control to assure that images in the MAPS database could be registered within acceptable tolerances of 2 to 4 pixels. However, given that the ground sample distance for many of the images is approximately one meter, and that MAPS contains a wide variety of imagery with difference ground scales, projections, from multiple sources, it is unlikely that one would be able to totally eliminate the initial registration error.

The calculated registration offset is then used to extract the Right image for a second time. The orthographic extraction process is given a new geographic box that has been translated by the calculated offset. In this way, the new Right image will be more nearly registered to the Left image than if we had simply translated the original Right image. Originally we felt the offset could be handled entirely within the SI stereo process and that resampling the Right image would be unnecessary. Experimentation showed this not to be the case, but since the internal offset capability was already added to SI, it is still in

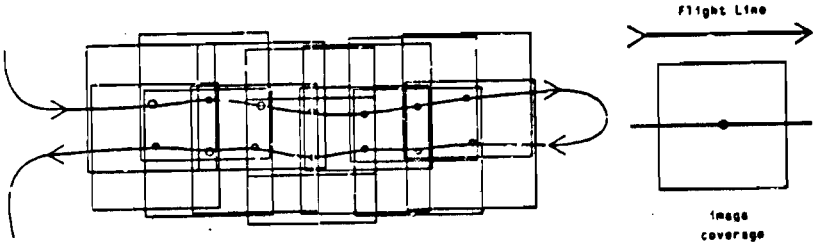


Figure 4-2: Mission Flight Lines

use. That is, even though we calculate an image offset and resample the Right image for a second time, we still later calculate any remaining offset between the Left and resampled Right images and use that value within the stereo process itself.

If necessary, the resulting Left-Right stereo pair images are rotated. The SI stereo process assumes that scanlines are stereo epipolar lines. Until this point the stereo pair scanlines were East-West. Earlier a rotation value was calculated for use in determining the size of the extraction area. The rotation value is the amount the images must be rotated to make the epipolar lines become scanlines and assure that the Left-Right pair create a positive stereo image (ie. tall objects shift inward). The rotation value is the baseline orientation that was calculated earlier as the angle at the geographic center of the original image between East and the line to the alternate image geographic center. After rotation, the appropriate subimage rectangle of real data is cropped from the rotated image since the rotation leaves four right triangles of non-data at the corners.

4.4. Run the Stereo Process

At this point all constraints required by the SI algorithm on the stereo pair have been met. The following few comments concern the specific use of the SI process.

The Left-Right stereo pair images are repeatedly smoothed to form the coarse-fine hierarchy of images used by SI. As in Section 4.3, SI again calculates a global registration offset value between the original Left image and the resampled Right image. This global offset is used internally by SI during its calculation of the disparity image. The disparity image result is saved for analysis upon completion of the SI disparity process.

4.5. Analyze the Results

In general the methods used in analyzing stereo results will depend on the stereo process used, the sensing method, and the type of disparity map produced by the process. Generally, one can characterize stereo matching results as one of the following:

- point correspondence(s)
- sparse depth map
- dense (complete) depth map

The objective of stereo verification is to determine if the region of interest has height relative to its surroundings. One of the major reasons for choosing SI as our stereo process is that its dense disparity image simplifies this analysis step. Analysis of sparse feature based depth results like those produced by edge-based or interest area-based stereo processes would require careful determination of whether a feature belongs to the region of interest or to its surroundings. One obvious method would be to interpolate the sparse depth results into a dense map similar to the SI disparity image. However it is not clear how reliable such a map would be, especially given the complex images presupposed in aerial interpretation, and techniques for doing such interpolation are still considered a topic for research[15]. The remainder of this Section describes how STEREOSYS analyzes SI disparity images and is illustrated by several examples.

In order to analyze the dense SI disparity image an overlaying bitmap of the region of interest is made. First the region's boundary point list is rectified to overlay the pre-rotated Left stereo pair image. The rectified boundary point list is then converted to a bitmap image of the region. Finally the bitmap is rotated to properly overlay the Left stereo pair image used in the disparity image calculation. The bitmap is used to distinguish the areas of the disparity image inside and outside the region of interest.

The disparity image and the overlaying region bitmap are used to calculate the mean and standard deviation for the disparity values of the areas within and without the region of interest. STEREOSYS uses a heuristic function that combines the standard deviations, S_{in} and S_{out} and the difference in the means, D, to determine four confidence values:

1. Overall Confidence in the Stereo Results.
2. Confidence in the Region having Little to No Height.
3. Confidence in the Region having Moderate Height.
4. Confidence in the Region having Significant Height.

The first measure describes the overall confidence that can be placed on the stereo results. The disparity image results can vary from excellent to useless due to limits in correcting for misregistration and from noise caused by nondescript areas (Section 5.1). The confidence in the result is calculated as an empirically weighted sum of the mean difference and standard deviations.

$$0.1D + 0.5S_{in} + 0.4S_{out}$$

The D term is further influenced by the disparity image contrast which is related to the disparity range. A very small range can decrease this term by an empirical factor of 0.2. The S_{out} term is further influenced by an estimate of the amount of expected height clutter in the area. If the area is expected to be cluttered with tall objects this term increases by an empirical factor of 0.75. Both the disparity range and clutter values are provided by the processing context that caused SPAM to invoke STEREOSYS. These contexts include rules that recognize situations where height information can disambiguate competing hypotheses as well as supply likelihoods of clutter and height.

Confidence values (2-4) measure whether the region of interest was found to fall in one of three disparity or height ranges, provided by SPAM. These measures are relative to the hypothesized disparity range, rather than absolute statements about the regions height. For example, "Little to No Height" could mean about 5 meters high if a very large disparity range was selected but could mean less than one meter if a small range was used.

All three height confidence values are based on the difference in the means, D, but can be influenced by the disparity range in a manner similar to the D term in the results confidence described above. These values reflect where the height of the region falls within the height range supplied by SPAM. Confidence (2) is maximized as D goes to zero. Confidence (3) is maximized when D is approximately $1/7$ of the full disparity range. Confidence (4) is maximized as D goes to maximum disparity. It should be remembered that very high objects can create disparity values beyond the range of maximum disparity in which case their extreme height would go unnoticed.

5. Experimental Results

This Section presents results produced by STEREOSYS that illustrate several of the important issues encountered during system development. We also amplify comments made in previous Sections concerning issues of registration, disparity estimates and automating the overall stereo process. It is important to keep in mind several issues regarding the SPAM task environment and these experiments. First, all of the aerial mapping photography in the MAPS image database is nominally vertical. Since each image is in "correspondence" with a ground control database, it is possible to compute the geographic coordinate for each pixel in the image. Of course, there are inherent inaccuracies in this process, both in measurement of the landmark positions and in recovering their position in the imagery. These inaccuracies lead to image offsets when the sub-image areas are extracted from the full image frames using geographic location. No assumptions are made with respect to the relative position of the cameras other than those described in Sections 4.1 and 4.2.

Second, the actual height of the region of interest is not calculated, that is, we do not solve the full camera equations, since the disparity image is scaled to record a particular range of elevations as describe in Section 4. One could calculate the baseline distance, and knowing the focal length and aircraft height, solve for the actual height, but given the statistical nature of our final analysis we have not found it to be necessary. Finally, the ground sample distance for the imagery reported on in this paper is approximately one meter per pixel.

Section 5.1 describes typical SI results before minor revisions were made to the matching algorithm and STEREOSYS was implemented. Section 5.2 illustrates the problem caused by database image misregistration and results produced by STEREOSYS. Section 5.3 deals with stereo pair preparation processes as well as how the SI results are analyzed. Section 5.4 describes test results from the automated use of STEREOSYS. Finally, Section 5.5 details a specific example from among the automated tests of Section 5.4.

Figure 5-1 shows one frame of aerial imagery containing National Airport in Washington, D.C. All of the examples in this paper come from various areas of this airport extracted from several stereo image pairs.

5.1. Preliminary SI Experiments

Before trying to build a stereo verification system using SI, we experimented with the overall process in order to get a feel for how SI might perform with MAPS images. Several issues arose: how to automatically set SI's initial disparity range values; deciding on modifications to provide the flexibility necessary to accommodate SPAM's requirements for a verification process; and how to analyze SI results. These first experiments were performed on stereo pair images registered by hand and extracted from the database using the same orthographic rectification process as in STEREOSYS.

Figure 5-2 shows the Left and Right stereo images of a long hangar building running diagonally from the top right to bottom left of each image. Below the Left image is the SI disparity image result. Within the disparity image dark areas are closer to the camera (higher) than are the light areas. The hangar is clearly shown to be higher than its surroundings. Some points of interest concerning the disparity image are:

- The speckled areas are caused by the loss of correspondence in large nondescript areas such as the large solidly shaded areas of pavement to the right and below the hangar. Such nondescript areas are characterized by the lack of edges or texture.
- Boundary edge effects show up as errors all around the disparity image. These effects are caused by lack of data outside the image and have been alleviated somewhat in the modified versions of SI.
- Stereo aliasing effects probably caused the problem with the curved hangar roof in the lower left corner. The white area in the roof indicates a concave section where none exists
- Temporal changes in the stereo pair images can cause unpredictable results. An example is the white area along the right side caused by the moving truck.

Figure 5-3 shows a taxiway/runway area of the airport. This area contains very little variation in height and contains large variations in image intensity. The disparity image shows no significant height for any image region but again illustrates the problems with large nondescript areas and edge effects. Note also that the edge effects are propagated into nondescript areas. The statistical analysis method described in Section 4.5 was chosen partially because of its ability to recognize these situations as not being a significant indication of elevation.

5.2. Registration Problem and Solution

The results of the previous Section were produced from stereo pairs that were registered by hand. That is, the identification of the extraction areas was not done automatically and any misregistration in the stereo pair was kept to less than two pixels. This can be contrast with the 6 to 15 pixel disparities we normally experienced in the images used of the Washington D.C. National Airport.

Through experimentation it was found that the SI process could sometimes produce fair results if the stereo pair was up to 6 pixels misregistered, but this was found to be far too restrictive for automatic purposes since the MAPS correspondence between database images can be off by as much as 30 pixels or more in areas with little ground control. Figure 5-4 shows early SI results from a stereo pair created automatically from the database.

The misregistration problem is handled by SI's ability to calculate a global disparity shift between pairs of images. STEREOSYS uses SI to calculate the shift between the originally extracted stereo pair then uses the shift value to re-extract the Right image. Figure 5-5 illustrates this process. The top two images are the original Left-Right stereo pair. The lower right contains the Right image after a calculated shift of 7 pixels vertical and 13 pixels horizontal has been eliminated.

Since the shift is an inexact statistical value SI was also modified to calculate and compensate for any remaining small misregistrations. The lower left of Figure 5-5 contains the disparity image that results from the combination of these techniques. This approach has demonstrated the ability to properly compensate for original misregistrations of up to 25 pixels. Beyond that point the global shift calculation normally fails. However this shortfall can be properly overcome by adding enough control to the image database to assure misregistrations will not exceed the limits of the registration process.

5.3. Analysis of Results

If the reader looked carefully at the stereo pair used in the last Section they might have noticed that the pair forms a negative stereo image. That is, objects with height lean away from one another and, if viewed in stereo, would form a reversed stereo image. In such an image buildings would appear to go down into the ground. To correct this, either the Left and Right images could be exchanged or both could be rotated 180 degrees. We choose to rotate the stereo pair images since this can be combined with the arbitrary rotations necessary to align scanlines and epipolar lines. The results in Figure 5-6 show the rotated stereo pair of Section 5.2. Note that the disparity images shown in Section 5.2 were produced after exchanging the images to from a positive stereo pair or else the disparity image would have shown negative height for the buildings.

In order to analyze the SI disparity images an overlaying bitmap of the region of interest is produced as described in Section 4.5. The bitmap is used to distinguish areas of the disparity image as being either inside or outside the region of interest. Based on this separation, the mean and standard deviation of disparity values within and without the region are calculated. STEREOSYS uses the standard deviations and the difference in the means in its heuristics that determine the stereo verification confidence values also described in Section 4.5. These values reflect confidence in the stereo result and confidence in the region of interest having little height, moderate height or significant height. The values are such that 0.0 signifies no confidence while 1.0 signifies "perfect" confidence. An example of the bitmap and the confidence values are also shown in Figure 5-6.

Figure 5-7 is an example of STEREOSYS results where the region of interest has no height

5.4. Fully Automatic Use

One important objective for STEREOSYS was that it be flexible enough to work reliably with all sorts of regions and in concert with SPAM. To test STEREOSYS against these goals, SPAM was given access to STEREOSYS for the purpose of stereo verification while trying to interpret the Washington D.C. National Airport area. STEREOSYS was called upon to give a verification analysis of 70 regions. Table 5-1 lists the confidence results for these regions of interest. The *Human Interpretation* column gives the correct interpretation for each region. The *SPAM Hypothesis* column gives the SPAM hypothesis used in invoking STEREOSYS. Exact interpretation of the *Low*, *Med* and *High* columns depends on what hypothesis SPAM had for the region when it invoked STEREOSYS. For example, if the hypothesis was for a low object like tarmac then *Low* would indicate a range in heights of 0-1 meters; *Med* 1-5 meters; and *High* 5-infinity. But, if the hypothesis was for a moderately tall object such as a hangar then *Low* would indicate a range of 0-5 meters; *Med* 5-12 meters; and *High* 12-infinity. A similar broadening of ranges would hold for very tall hypothesis but in this case of airport analysis, such hypothesis are not used.

Close examination of Table 5-1 reveals that as the *Result* confidence decreases height confidences tend to move toward *Low*. This is because disparity images with low *Result* confidences are randomly noisy. This causes the mean values for the areas within and without the region of interest to become nearly equal. The heuristics calculating height confidences rely mostly on the difference in these means; no difference indicates no height. The very few cases where poor results cause confidence in the region being tall happen when the region of interest is very small and happens to lie on a random dark area of the disparity image.

Table 5-2 summarizes the test by categorizing result confidence values. This data primarily reflects how often the system was able to properly register the stereo pair. Result confidences of over 0.6 (out of 1.0) reflect good registration. Result confidences below 0.4 reflect cases where the system was probably unable to determine the shifts necessary to bring the stereo pair into registration. Values 0.4 - 0.6 can be caused by areas cluttered with high objects, highly nondescript areas or registration problems. Poor results due to bad registration can be alleviated through the addition of correspondence control between the data base images to achieve a better initial camera model. However, this is unlikely to be a viable solution in practice due to the expense of adding ground control points. The remaining problems like nondescript areas and moving objects are inherent in the stereo process itself and are not dealt with in this work. Table 5-2 also summarizes how well the confidence results agreed with human height evaluation for the regions being verified. For the purposes of this evaluation a "winner take all" strategy is used. That is, the height confidence range having the highest confidence was deemed to be the height assigned to the region by STEREOSYS.

The careful reader will notice that the "% Human Agreement" value in Table 5-2 does not decrease when the "Confidence" value is below 0.4. The disparity image results for regions with such low confidence

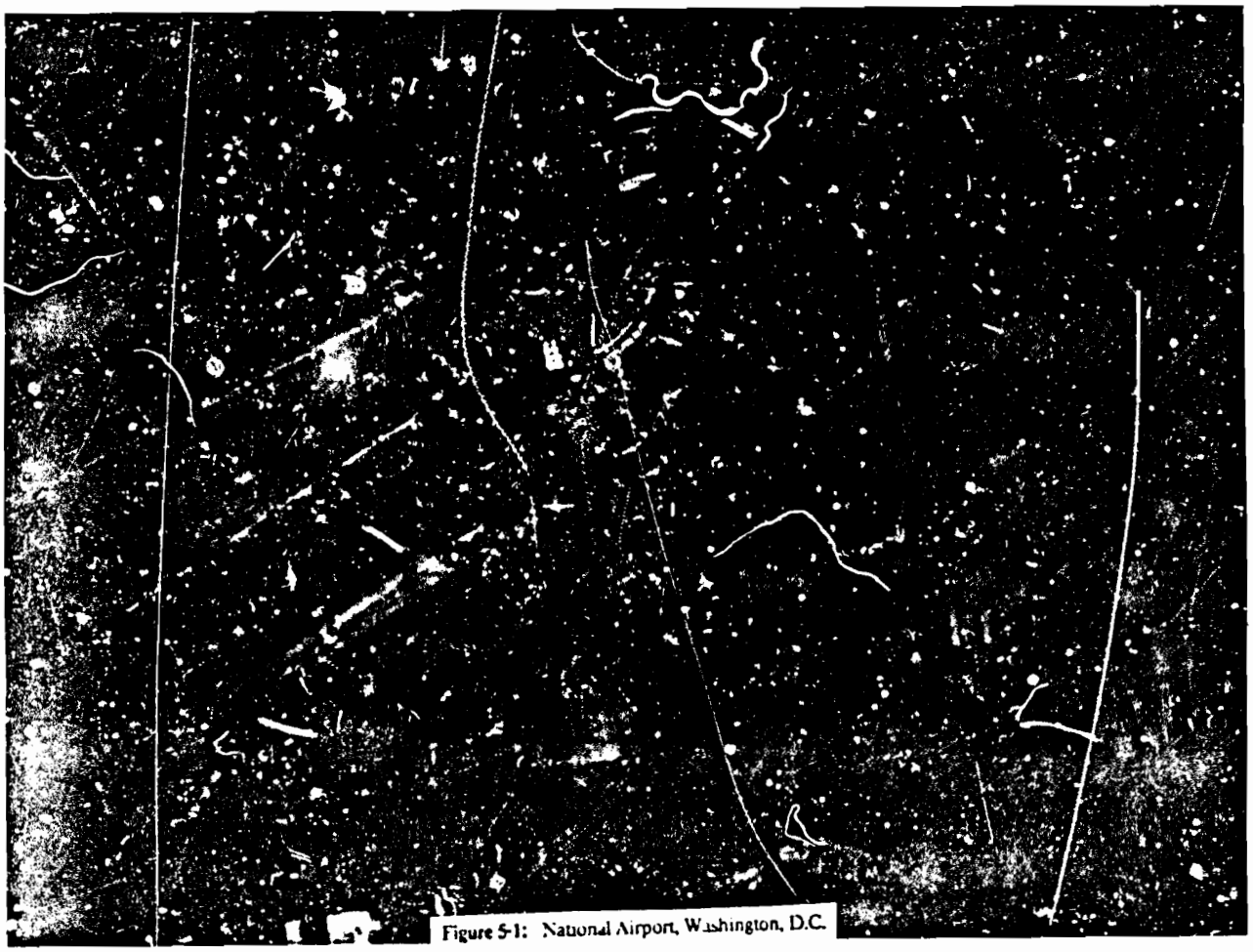


Figure 5-1: National Airport, Washington, D.C.

are usually randomly noisy. Statistically, this causes the difference in mean disparity values between the areas inside and outside the region to approach zero which, in turn causes the STEREOSYS height confidence heuristic to favor a low height interpretation. Of the 70 test cases presented, only eight are of regions with significant height and of these, only one created a result confidence below 0.4. Since the remaining poor result confidences were caused by regions without height, and STEREOSYS favors a low height interpretation in cases where it can't calculate an answer, this somewhat inflates the percent of human agreement within the low confidence range.

Tables 5-3 through 5-6 are confusion matrices showing the performance of STEREOSYS over several result confidence ranges. The table columns are the number of times a SPAM hypothesis was correct or incorrect, with respect to height, as compared to a human interpretation. For example, if SPAM hypothesized a low height region, say grassy area, and the region was any other low height region, say tarmac, then the hypothesis was deemed correct. The table rows indicate the number of times STEREOSYS confirmed or rejected the SPAM hypothesis. A perfect result would find STEREOSYS always confirming correct hypotheses and rejecting the incorrect hypotheses, ie., zeros in the lower left and upper right elements of the confusion matrix.

Table 5-7 indicates that STEREOSYS performs well with objects having height. One initial concern with the S1 stereo process is that often, when it is initiated with too small a disparity range, the S1 method will not converge to a useful result. This could be the case when SPAM hypothesizes an object with no height and in reality the object has significant height. To lessen the chance of this problem occurring we tried to be generous in the size of our three standard height ranges. In the one case where this situation actually occurred, STEREOSYS produced the correct response.

5.5. A Detailed Example

As an example of how stereo verification can aid image analysis, this Section describes one of the 70 invocations of STEREOSYS by SPAM from Section 5.4. This Section is included also to give the reader a flavor for how SPAM, a rule-based production system, utilizes stereo verification. As mentioned in the introduction, SPAM may invoke STEROSYS in one of two modes. During early stages of scene interpretation SPAM gathers low-level information by testing newly generated regions for height in order to develop an initial set of hypotheses for the region. During later processing, as collections of regions begin to be combined into components of the airport model, STEREOSYS is employed to disambiguate between two or more plausible but conflicting hypotheses. This Section describes the former situation by showing extracts from the SPAM and STEREOSYS execution traces. These extracts have been edited slightly to enhance their readability.

Figure 5-8 contains several of the SPAM OPS5[16] rules that lead to the invocation of STEREOSYS. The first rule, region-to-fragment::get-depth, is used to recognize an appropriate point for the invocation of STEREOSYS. The firing of this rule causes SPAM to change its operating *context* to the *generate-depth-info task*. The next two rules are examples of rules activated by this context. They will set up the STEREOSYS parameters appropriate for the region of interest's current best hypothesis based on an assigned confidence value. These

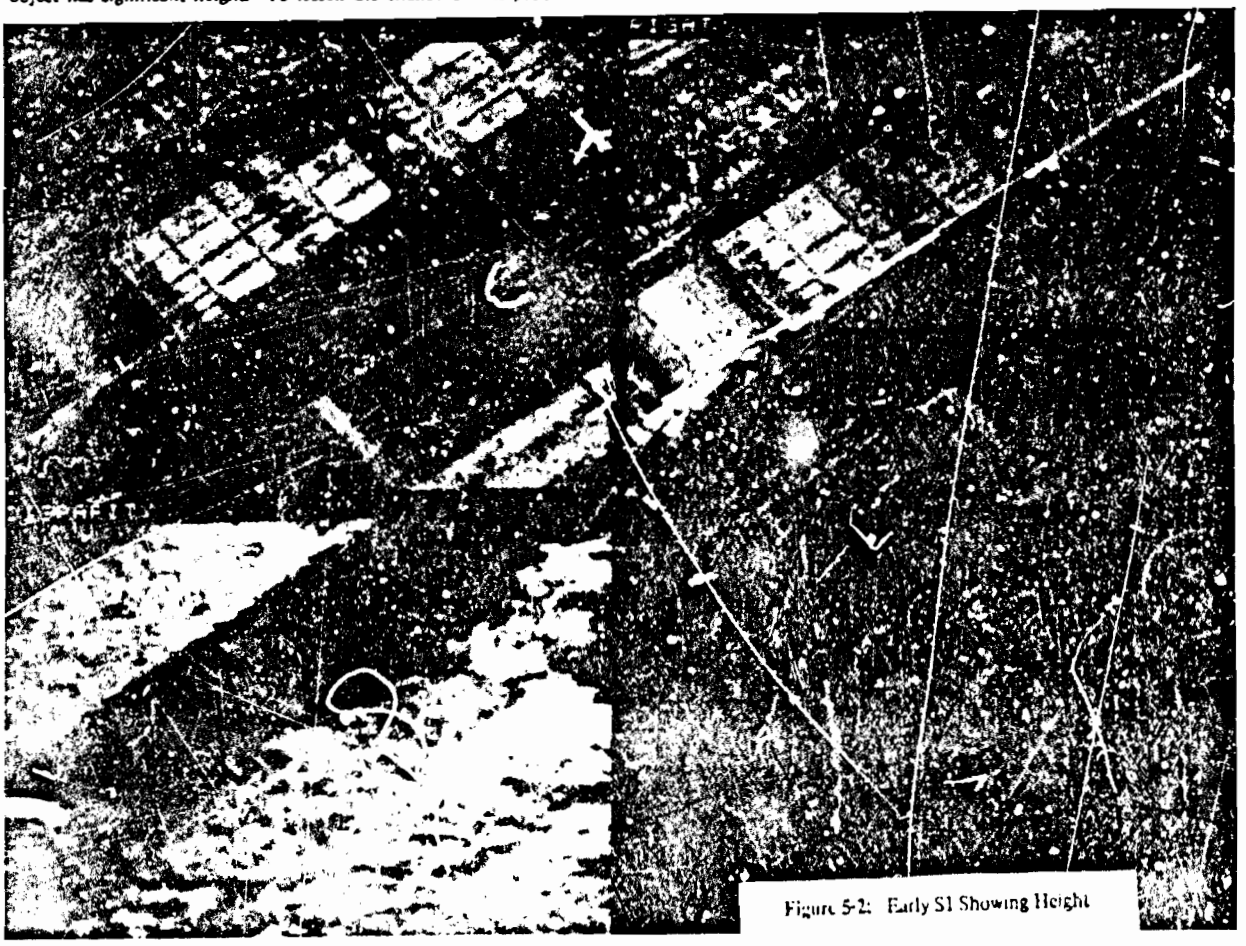


Figure 5-2: Early S1 Showing Height

REG ID	Human Interpret	SPAM Hypothesis	Confidences: Result	Low	Med	High
R01	Runway	Runway	0.86	0.98	0.02	0.00
R02	Runway	Runway	0.42	0.98	0.02	0.00
R03	Runway	Runway	0.33	0.98	0.02	0.00
R04	Parking-apron	Hanger	0.34	0.98	0.02	0.00
R05	Grassy-area	Parking-apro	0.28	0.42	0.48	0.10
R06	Runway	Runway	0.44	0.21	0.60	0.19
R07	Runway	Runway	0.40	0.82	0.16	0.02
R08	Taxiway	Parking-apro	0.35	0.30	0.57	0.13
R09	Taxiway	Hanger	0.56	0.15	0.57	0.30
R10	Taxiway	Hanger	0.29	0.98	0.02	0.00
R12	Taxiway	Parking-apro	0.69	0.70	0.25	0.06
R13	Taxiway	Hanger	0.58	0.51	0.41	0.08
R14	Taxiway	Hanger	0.23	0.75	0.22	0.03
R15	Taxiway	Hanger	0.30	0.98	0.02	0.00
R16	Taxiway	Hanger	0.37	0.98	0.02	0.00
R17	Taxiway	Hanger	0.78	0.22	0.60	0.18
R18	Taxiway	Parking-lot	0.32	0.98	0.02	0.00
R19	Taxiway	Terminal	0.27	0.98	0.02	0.00
R20	Grassy-area	Hanger	0.32	0.67	0.28	0.0
R22	Grassy-area	Parking-apro	0.46	0.43	0.48	0.09
R23	Grassy-area	Hanger	0.21	0.19	0.60	0.21
R25	Grassy-area	Hanger	0.31	0.98	0.02	0.00
R26	Grassy-area	Parking-apro	0.40	0.98	0.02	0.00
R27	Grassy-area	Hanger	0.94	0.55	0.36	0.07
R28	Grassy-area	Parking-apro	0.40	0.50	0.42	0.08
R29	Grassy-area	Hanger	0.30	0.27	0.68	0.18
R30	Grassy-area	Hanger	0.67	0.98	0.02	0.00
R32	Grassy-area	Hanger	0.73	0.98	0.02	0.00
R35	Grassy-area	Parking-apro	0.47	0.98	0.02	0.00
R36	Grassy-area	Hanger	0.57	0.98	0.02	0.00
R37	Grassy-area	Hanger	0.66	0.57	0.36	0.07
R38	Grassy-area	Parking-apro	0.26	0.39	0.51	0.10
R40	Grassy-area	Hanger	0.62	0.89	0.10	0.01
R41	Parking-lot	Parking-apro	0.20	0.98	0.02	0.00
R43	Parking-lot	Runway	0.48	0.98	0.02	0.00
R44	Parking-lot	Hanger	0.86	0.98	0.02	0.00
R45	Parking-lot	Parking-apro	0.48	0.36	0.53	0.11
R46	Parking-lot	Parking-apro	0.75	0.18	0.69	0.28
R47	Parking-lot	Parking-apro	0.29	0.98	0.02	0.00
R48	Parking-lot	Parking-apro	0.26	0.24	0.59	0.17
R49	Parking-lot	Hanger	0.81	0.60	0.42	0.08
R50	Grassy-area	Grassy-area	0.27	0.98	0.02	0.00
R51	Hanger	Hanger	0.68	0.18	0.18	0.79
R52	Terminal	Hanger	0.42	0.05	0.18	0.79
R53	Hanger	Hanger	0.93	0.09	0.47	0.44
R54	Grassy-area	Parking-apro	0.78	0.60	0.34	0.06
R55	Taxiway	Hanger	0.22	0.57	0.37	0.08
R56	Taxiway	Terminal	0.30	0.41	0.49	0.10
R57	Hanger	Parking-apro	0.58	0.06	0.18	0.78
R58	Hanger	Hanger	0.26	0.74	0.23	0.03
R62	Parking-lot	Hanger	0.85	0.98	0.02	0.00
R64	Grassy-area	Parking-apro	0.40	0.12	0.56	0.32
R65	Grassy-area	Parking-apro	0.39	0.51	0.41	0.08
R66	Grassy-area	Grassy-area	0.36	0.27	0.58	0.15
R67	Grassy-area	Parking-apro	0.49	0.98	0.02	0.00
R68	Grassy-area	Parking-area	0.39	0.65	0.29	0.06
R73	Terminal	Hanger	0.87	0.10	0.52	0.38
R75	Terminal	Hanger	0.65	0.09	0.50	0.41
R77	Terminal	Hanger	0.60	0.07	0.39	0.54
R78	Tarmac	Parking-apro	0.69	0.98	0.02	0.00
R79	Parking-apro	Parking-apro	0.43	0.95	0.05	0.00
R80	Parking-apro	Parking-apro	0.36	0.54	0.38	0.08
R81	Tarmac	Parking-apro	0.24	0.98	0.02	0.00
R82	Parking-apro	Grassy-area	0.50	0.98	0.02	0.00
R83	Tarmac	Parking-apro	0.55	0.20	0.59	0.21
R84	Parking-apro	Hanger	0.39	0.98	0.02	0.00
R85	Tarmac	Parking-apro	0.17	0.91	0.08	0.01
R87	Road	Taxiway	0.29	0.07	0.36	0.57
R94	Road	Road	0.48	0.14	0.57	0.29
R95	Road	Runway	0.27	0.98	0.02	0.00

Table 5-1: Test Results

Confidence	% of Tests	% Human Agreement
0.6-1.00	27.1	89.5
0.5-.599	8.6	66.7
0.4-.499	20.0	64.3
0.0-.3.9	44.3	67.3

Table 5-2: Test Summary

Hypothesis	Correct	Incorrect
Confirmed	31	5
Rejected	14	20

Table 5-3: All Result Confidences [0.0 - 1.0]

Hypothesis	Correct	Incorrect
Confirmed	[illegible]	[illegible]
Rejected	7	12

Table 5-4: Result Confidences [0.4 - 1.0]

Hypothesis	Correct	Incorrect
Confirmed	10	1
Rejected	2	12

Table 5-5: Result Confidences [0.5 - 1.0]

Hypothesis	Correct	Incorrect
Confirmed	9	0
Rejected	1	9

Table 5-6: Result Confidences [0.6 - 1.0]

Hypothesis	Correct	Incorrect
Confirmed	6	0
Rejected	0	1

Table 5-7: Regions with Actual Height [0.4 - 1.0]

parameters are an indication of the expected height range and height clutter for the region of interest. The rules for setting [illegible] parameters appropriate for a runway or a hangar hypothesis are shown. The rule applicable to the hypothesis with the highest confidence value will fire. Along with setting the necessary parameters, the rule firing will change the context to *get-depth* in order to fire the next rule, specific::get-region-depth, which actually invokes the STEREOSYS process.

Figure 5-9 is an excerpt from the SPAM trace just before it invoked STEREOSYS. The region of interest is Hand36809-N.37 – 0 ("R37" for short). Rule firings 853 to 856 step through the development of hypothesis confidences for region R37. By the end of this sequence of rules the region had a 0.94 confidence of being a hangar and a 0.68 confidence of being a grassy area. These interpretations were based on weak heuristics and measurements such as 2D shape and texture. SPAM also uses knowledge concerning the spatial consistency of this hypothesis with other region hypotheses in the airport scene to evaluate confidence, but this knowledge is applied after an initial assignment of plausible hypotheses based upon these simple measures[2]. In order to avoid a combinatorial explosion of hypotheses at the spatial consistency phase, it is important that STEREOSYS be able to refute the incorrect hangar hypothesis.

Rule firing 857 changed the operation context to the get-depth task because the hangar hypothesis had the highest confidence of any interpretation for this region. The get-depth task rule is used to set up the invocation of the STEREOSYS process by SPAM. Based on the type of hypothesis and other knowledge about airport organization, SPAM selects parameters for height-range and clutter to be used during rule firing 858. Finally, using these parameters set for finding height information about typical hangars, STEREOSYS was invoked.

```
(p region-to-fragment::get-depth
  { <context> (context ^task region-to-fragment
       ^datum <token>) }
  (region ^token <token> ^region-status interpreted)
- (context ^task <> region-to-fragment)
- (store-results ^result-one class-match)
- (store-results ^result-one subclass-match)
--> (remove <context>)
    (make context ^task generate-depth-info
        ^datum <token>))

(p depth::get-runway-depth
  { (context ^task generate-depth-info
       ^datum <token>) <context> }
  (fragment ^region-token <token> ^hypothesis runway
       ^confidence <c>)
- (fragment ^region-token <token>
       ^hypothesis <> unknown ^confidence > <c>)
- (context ^task <> generate-depth-info)
--> (remove <context>)
    (bind <height-estimate> 0-6)
    (bind <cluttering> isolated)
    (make context ^task get depth
      ^datum <token> <height-estimate>
      <cluttering> runway))
```

```
(p depth::get-hangar-depth
  { (context ^task generate-depth-info
       ^datum <token>) <context> }
  (fragment ^region-token <token> ^hypothesis hangar
       ^confidence <c>)
- (fragment ^region-token <token> ^hypothesis <> unknow
       ^confidence > <c>)
- (context ^task <> generate-depth-info)
--> (remove <context>)
    (bind <height-estimate> 10-inf)
    (bind <cluttering> cluttered)
    (make context ^task get-depth
      ^datum <token> <height-estimate> <cluttering> han

(p specific::get-region-depth
  { (context ^task get-depth
  ^datum <regtok> <height-est> <regcontext>
    <hyp>) <co text> }
  (global-status ^current-image <img>)
  (interp-constants ^output-file <outfile>)
  { <region> (region
      ^token <regtok> ^symbolic-name <symname>) }
--> (call depth <symname> <img> AREAL <height-est>
          <regcontext> -o <outfile>)
    (remove <context>)
    (modify <region> ^depth-low <lowdepth>
       ^depth-moderate <moddepth> ^depth-high <highdept
```

Figure 5-8: OPS5 Production Rules

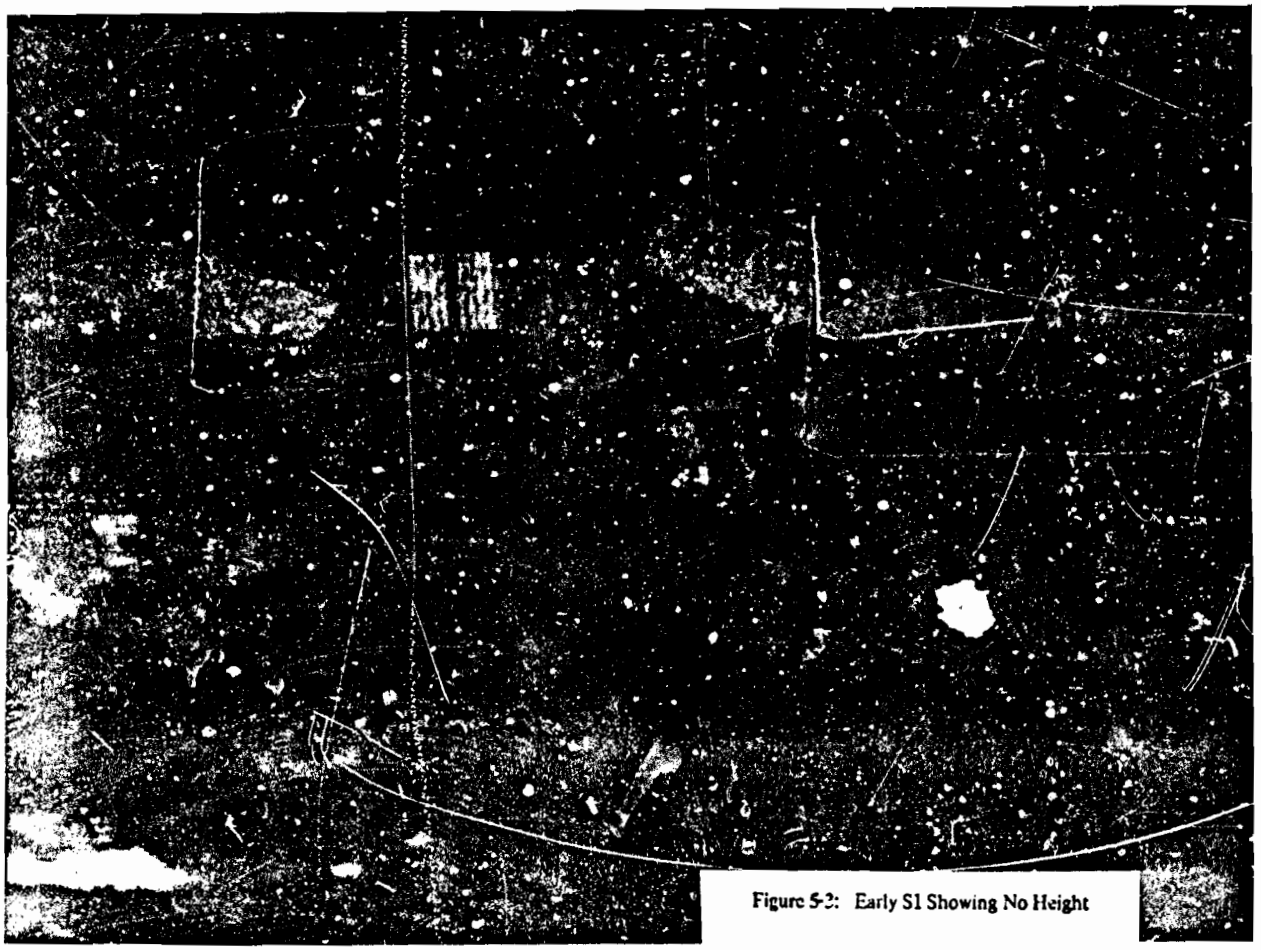


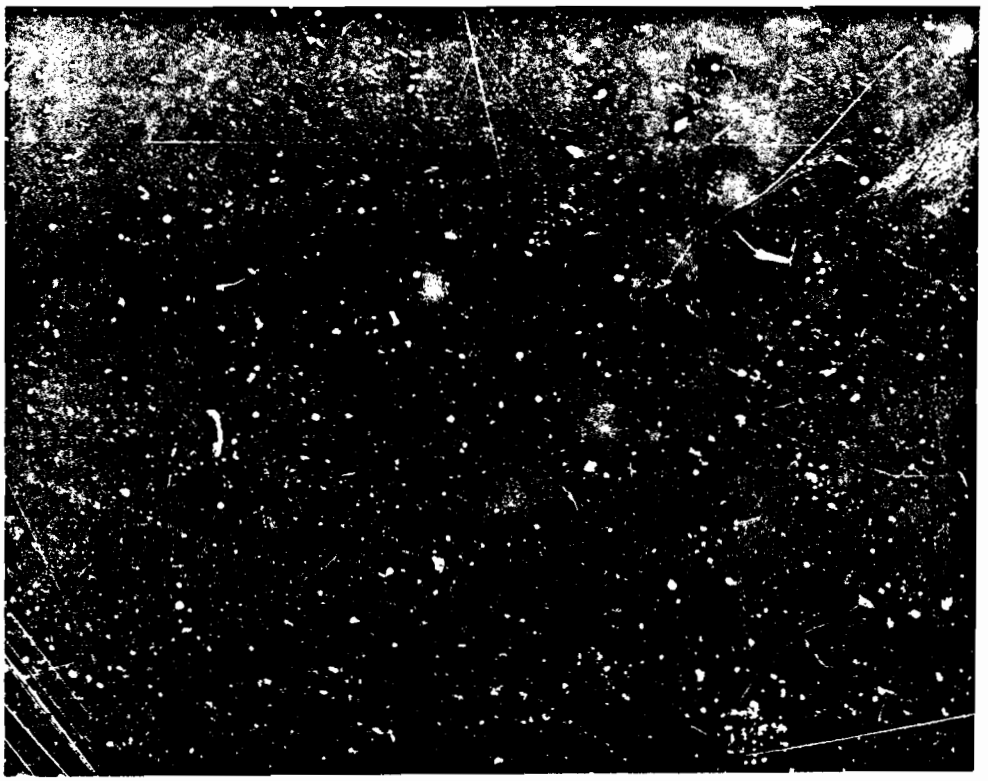
Figure 5-3: Early S1 Showing No Height

Figure 5-4: Early S1 against Poor Registration

Figure 5-5: Misregistration Solution

Result Confidence	0.651
Low Height Confidence	0.051
Moderate Height	0.159
Significant Height	0.791

Figure 5-6: Analysis Results Indicating Height



Result Confidence	0.994
Low Height Confidence	0.963
Moderate Height	0.039
Significant Height	0.000

Figure 5-7: Analysis Results Indicating No Height

```
853. region-to-fragment::generate-subclass-match 2939
   2912 2938 CONFID-LIST  (1.0 0 909938 0.0 .853321)
Match of hangar region Hand36809-N.37_0 = .9408
854. interpret-as-hangar 2946 1? 2912 2935 2938
        Interpreting region Hand36809-N.37_0
855. region-to-fragment::generate-subclass-match 2923
   2912 2976 CONFID-LIST  (1.0 .918012  124357)
Match of grassy-area  region Hand36809-N.37_0 = .8807
856. interpret-as-grassy-area 2978 12 2912 2974 2378
        Interpreting region Hand36809-N.37_0
857. region-to-fragment::get-depth 2912 2984
858. depth::get-hangar-depth 2986 2984
859. specific::get-region-depth 2988 2982 2 2984
```

Figure 5-9: SPAM Execution

```
A: STEREOSYS: Region_id     = Hand36809-N.37_0
              Generic       = dc36809
              Region_type   = AREAL
              Height_range  = 10-inf
              Clutter       = cluttered
   STEREOSYS: Region Temp file key is R09-37
B:   D3_INFO:
     D3_file = /visf/airport/Hand36809-N/37A0.d3
     X = 139841.816519  Y = 277348.279196
C: STEREOSYS:
   d3entcor /visf/airport/Hand36809-N/37A0 d3 R09-37.ec
D: STEREOSYS: sorting EC file by stereo coverage
   STEREOSYS: selecting best coverage:
        Left       = dc36809      ScaleL = 0.000083
        Right      = dc36808      ScaleR = 0.000083
        HalfHeight = 82   HalfWidth = 44
        Rotation = 0.782102
        Del_lat  = 0.040686  Del_lon  = 0.053418
E: STEREOSYS:
   ortho dc36809 38 50 34 902 77 2 23 472
                 38 50 48 330 77 2 33 85
                 -m 1.056782 R09-37.l.tmp
   Mapping lbw image of dc36809 to the box formed by:
        lat N38 50 34 (902)    lon W77 2 23 (472)
   and  lat N38 50 48 (330)    lon W77 2 33 (88)
   Requested gridsize:  1.06 meters
   Actual gridsize: .0407x.0528 sec. (1.05x1.06 meters)
   Size of result: 60390 bytes (330 rows X 183 columns)
   STEREOSYS:
   ortho dc36808 38 50 34 902 77 2 23 472
                 38 50 48 330 77 2 33 86
                 -m 1.056782 R09-37.r.pre
F: STEREOSYS: Off Set cmd file created = b25825.off.tmp
   STEREOSYS: s1 b25825.off.tmp
G: STEREOSYS:
   ortho dc36808 38 50 34 374 77 2 22 741
                 38 50 47 802 77 2 32 384
                 -m 1.056782 R09-37.r.tmp
   STEREOSYS: mv R09-37.r.tmp R09-37.right
   STEREOSYS: mv R09-37.l.tmp R09-37.left
H: STEREOSYS: S1 command file created = c25825.cmd.tmp
   STEREOSYS: s1 c25825.cmd.tmp
I: STEREOSYS: Created warped SEG file  R09-37.w.seg
   STEREOSYS:
   segtoimg R09-37.w.seg -o R09-37.b.tmp -I R09-37.left
   STEREOSYS: mv R09-37.b.tmp R09-37.bitmap
J: STEREOSYS: Stereo statistics for Hand36809-N.37_0
     Mean difference:  5.41722
     Region stddev:  26.9966
     Backgnd stddev:  34.3314
     Result confidence:  0.661238
     Low depth confidence:  0.673924
     Moderate depth confidence:  0.350413
     High depth confidence:  0.085663
```

Figure 5-10: STEREOSYS Execution

Figure 5-10 gives extracts from the STEREOSYS trace of region R37. An explanation of the trace, coded by the bold capital letters, follows:

- A: The input parameters are listed. The parameter *Height-range* determines what disparity range S1 will use. In this case it was set to *10-inf* because SPAM thought R37 might be a hangar which is usually 10 or more meters high.
- B: The database is used to find the boundary list file and centroid for R37.
- C: Again the database is used to create an unsorted ".ec" coverage file of images containing R37.
- D: The coverage file is sorted; the best images are selected; and the extraction regions are calculated. Notice that since the rotation value is so near zero the later rotation steps are skipped over and replaced by simple UNIX moves (mv).
- E: The Ortho process extracts an orthographically rectified stereo pair.
- F: S1 is invoked to calculate any misregistration between the pair. The calculated offset is not shown in the trace, but for this particular region it was 13 pixels vertically and 14 pixels horizontally.
- G: Ortho is called to extract a new Right image based on the calculated offset.
- H: S1 is again invoked to calculate any remaining offset and produce a disparity image.
- I: The boundary list for *R37* is warped, or rectified, to overlay the Left image extraction area. The result is a ".seg" file which is converted to a bitmap.
- J: The disparity image is analyzed; the statistics are converted into confidence values and the results are sent back to SPAM.

These particular results are interesting in that SPAM sent R37 to STEREOSYS with a current hypothesis that R37 was a hangar but got back a fairly confident indication that there was no appreciable height present. That is, the result confidence was 0.66 with a 0.57 confidence that R37 had little to no height. Figure 5-11 shows R37's originally extracted stereo pair, the disparity result and region bitmap.

The effect of temporal change and the relative insensitivity of STEREOSYS to such changes is evident in this experiment. Notice that the plane on the taxiway in the Right image has moved a significant distance toward the top of the area of interest in the Left image. This causes two artifacts in the disparity image calculated by STEREOSYS. The area in the Left image where one would have expected to find a plane corresponding to the one in the Right image shows a very large disparity (height) estimate. Similarly, the missing plane in the Right image shows a very small (nearly zero) disparity estimate. Since we are looking at global statistics over the entire the region bitmap and the surrounding area of interest, these anomalies represent a small portion of the statistical sample. It is probably the case that the result and height confidence measures would have been a little better had this situation not occurred. However, small temperal changes can be expected to occur, and it is important for the verification system to be relatively insensitive to them.

6. Conclusions

We believe that using height information in verification of aerial image analysis is an important approach and that the general stereo verification steps of Section 4 are minimal and applicable to all image analysis supported by an image database. In this context, our work with STEREOSYS has explored the pertinent issues and found viable solutions to the following important questions:

1. How can an aerial image database automatically generate a useful stereo pair containing an arbitrary region?
2. How can a stereo system handle the misregistration problems inherent in variable sourced image databases?
3. What kind of stereo results are appropriate for use in a verification process?
4. How can stereo results be analyzed so as to reflect not only the presence (or absence) of height but also the inherent reliability of the results?

STEREOSYS is not an infallible stereo verification system as indicated by the experimental results presented in Section 5. However, STEREOSYS is a highly flexible system that accomplishes the *entire* stereo process *automatically* from selecting image coverage to analyzing the stereo results while using an image database that has less than perfect image correspondence capabilities. From this viewpoint, we feel STEREOSYS has demonstrated the potential use of stereo verification in aerial image analysis.

If one defines stereo verification, as we do, to be a process whose purpose is to give a simple indication of the depth of one region in an image relative to the rest of the image, then stereo verification can be seen to be applicable to any domain where the identification of regions with significant differences in depth is important. For example, stereo verification could be useful for collision avoidance in mobile robotics or for the initial locating of tall objects in aerial photographs. This is especially true if an emphasis is placed on the use of fast and flexible processes. STEREOSYS has shown itself to be flexible but lacking in speed primarily due to the necessity for subimage rectification during the extraction of the stereo pair images. The registration step, needed to determine the offset in the originally extracted pair, is also time consuming. Approximate time for each experiment is about 20 cpu minutes using a VAX 11/780 under the UNIX operating system. However, we believe stereo verification can be done far more efficiently, particularly by using specialized hardware whose architecture is tailored to the matching and rectification algorithms. If so, this method can be a powerful component of a knowledge-based image analysis system and can greatly improve its ability to generate an accurate scene description.

Many different approaches to performing passive photographic stereo have been studied[17] and several have been implemented but few have been incorporated into systems that accomplish anything useful beyond producing aesthetic results if given a tightly controlled stereo pair. Flexible stereo verification is a useful application of stereo processes. Our work has outlined the general process of stereo verification and has studied how one stereo process, SI, can do useful

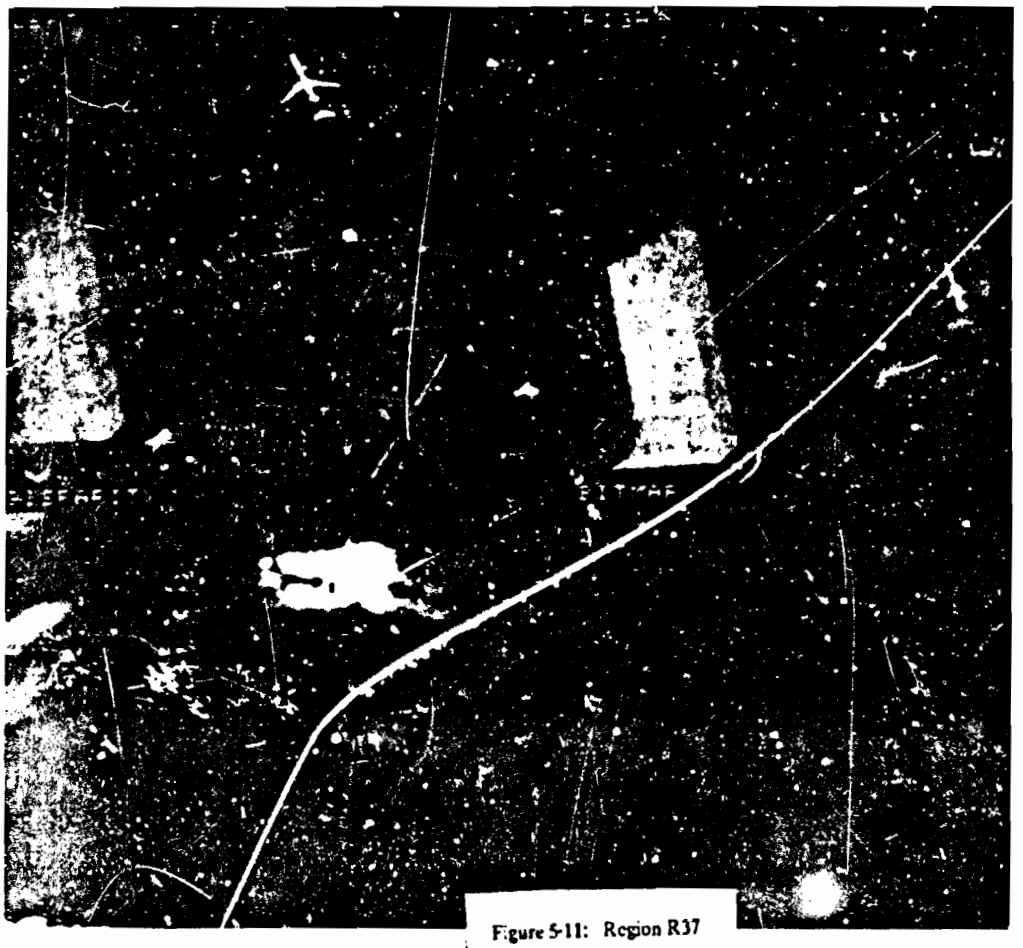


Figure 5-11: Region R37

verification work. We believe that one immediate direction of study in stereo verification should be in the testing of other known stereo processes in stereo verification systems.

7. Acknowledgements

Wilson Harvey integrated STEREOSYS into the SPAM rule-based aerial photo interpretation system and enabled us to run experiments using realistic system hypotheses and machine generated segmentations. Steven Shafer and John McDermott commented extensively on an earlier draft of this paper. Gudrun Klinker and Victor Milenkovic provided additional helpful comments.

Clifford McVay was a Visiting Researcher to the CMU-CSD during the academic year 1984-1985 from the Defense Mapping Agency Aerospace Center in St. Louis (DMAAC). Bruce Lucas is currently a System Designer in the Information Technology Center (ITC) at Carnegie-Mellon working on computer graphics and user interfaces. David McKeown is currently a Senior Project Scientist in the Computer Science Department working in Digital Mapping and Aerial Photo Interpretation.

This research was partially sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory Under Contract F33615-81-K-1539 and by the Defense Mapping Agency Under Contract DMA 800-85-C-0009. 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, of the Defense Mapping Agency, or the U.S. Government.

8. References

1. McKeown, D.M., and McDermott, J., "Toward Expert Systems for Photo Interpretation", *IEEE Trends and Applications '83*, May 1983, pp. 33-39.

2. McKeown, D.M., Harvey, W.A. and McDermott, J., "Rule Based Interpretation of Aerial Imagery", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 5, September 1985, pp. 570-585.

3. McKeown, D.M. and Pane, J. F., "Alignment and Connection of Fragmented Linear Features in Aerial Imagery", *Proceedings IEEE Computer Vision and Pattern Recognition Conference*, June 1985, Also available as Technical Report CMU-CS-85-122

4. McKeown, D.M., Denlinger, J.L., "Map-Guided Feature Extraction from Aerial Imagery", *Proceedings of Second IEEE Computer Society Workshop on Computer Vision: Representation and Control*, May 1984, Also available as Technical Report CMU-CS-84-117

5. McKeown, D.M., "MAPS: The Organization of a Spatial Database System Using Imagery, Terrain, and Map Data", *Proceedings: DARPA Image Understanding Workshop*, June 1983, pp. 105-127, Also available as Technical Report CMU-CS-83-136

6. McKeown, D.M., "Digital Cartography and Photo Interpretation from a Database Viewpoint", in *New Applications of Databases*, Gargarin, G. and Golembe, E., ed., Academic Press, New York, N. Y., 1984, pp. 19-42.

7. Lucas, B. D., *Generalized Image Matching by the Method of Differences*, PhD dissertation, Computer Science Department, Carnegie-Mellon University, July 1984.

8. Moravec, H. P., *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*, PhD dissertation, Computer Science Department, Stanford University, September 1980.

9. Baker, H. H., "Depth from Edge and Intensity Based Stereo", AIM 347, Stanford Artif. Intell. Lab, 1981.

10. Ohta, Y and Kanade, T., "Stereo by Intra- and Inter-scanline Search Using Dynamic Programming", *IEEE PAMI*, Vol. PAMI-7, No. 2, March 1985, pp. 139-154.

11. Herman, M., Kanade, T., and Kuroe, S., "Incremental Acquisition of a Three-Dimensional Scene Model from Images", *IEEE-PAMI*, Vol. PAMI-6, No. 3, May 1984, pp. 331-340.

12. Panton, D. J., "A Flexible Approach to Digital Stereo Mapping", *Photogramm. Eng. Remote Sensing 44*, December 1978, pp. 1499-1512.

13. Henderson, R. L., Miller, W. J., Grosch, C. B., "Automatic Stereo Reconstruction of Man-made Targets", *Computer Vision, Graphics, and Image Processing*, Vol. 1861979, pp. .

14. Vocar, J.M. and Faiss, R. O., "Image Magnification On STARAN", Tech. report GER-16342, Goodyear Aerospace Corporation, August 1976.

15. Terzopoulos, D., "Multi-Resolution Computation of Visible-Surface Representations", Tech report, Dept. of EE & CS., MIT, 1983, Ph.D Thesis

16. Forgy, C. L., "The OPS5 User's Manual", Tech. report, Carnegie-Mellon University, Department of Computer Science, 1981.

17. Barnard, S. T. and Fischler, M. A., "Computational Stereo", *Computing Surveys*, Vol. 14, No. 4, December 1982, pp. 553-572.

GC970145-36

THE TERRAIN-CALC SYSTEM

Lynn H. Quam, Artificial Intelligence Center

SRI International
333 Ravenswood Avenue, Menlo Park, California 94025

OVERVIEW

Terrain-Calc is a system for synthesizing realistic sequences of perspective views of real-world terrain that is described by a database consisting of geometric and photometric models. The geometry of the surface is described by a digital terrain model, which is a 2-dimensional array of elevations defined on a regular grid. The photometry of the terrain is described by a source image covering all or part of the area contained in the terrain model. This image is geometrically related to the terrain model by a projection (usually a perspective projection) that relates world coordinates to image coordinates.

The image-synthesis process is approximately equivalent to the following physical analogue:

1. Create a physical model of the terrain using a construction material that has a Lambertian reflectance function.

2. Project the source image onto the terrain model using a projector with proper focal length, placed at the proper position and orientation (equivalent to the perspective projection model relating the source image to the terrain).

3. View the physical terrain model with a camera having the desired focal length, position, and orientation.

Views constructed according to this description are approximately what would have been seen by a camera as defined by (3) over the actual terrain at the same time that the source image was acquired. The differences are due to the following effects:

- The geometric and photometric models are limited in resolution and accuracy.

- Portions of the surface that should be visible in the synthesized view were not visible in the source image.

- The actual surface materials do not obey Lambert's Law.

The view-synthesis algorithm is related to a technique developed by the computer graphics community called "texture mapping" (Quam 1971), (Blinn 1978), (Catmull 1980). A novel algorithm is used in Terrain-Calc to avoid aliasing that results from violating the sampling theorem.

THE MODELS

The geometry of the surface is described by a digital terrain model that is a 2-dimensional array of elevations $z(x,y)$, where x and y are defined on a regular grid. Each square of the grid is cut into two planar triangular facets, choosing the diagonal that maximizes the angle between the normals to the two triangular facets.

The photometry of the terrain is defined by a digitized source image covering all or part of the geometric model. It is assumed that the surface materials obey Lambert's Law, which makes it possible to generate relatively realistic views without detailed modeling of the surface materials.

The relationship between digitized pixel values in the source image and real-world luminous flux at the surface is generally unknown because of the many parameters in the film processing chain before the image is digitized and because of the effects of atmospheric scattering. For images acquired with calibrated sensors, it would be possible to synthesize views where the light source is at a position different from that in the source image, so long as the terrain obeys Lambert's law.

THE VIEW-SYNTHESIS ALGORITHM

Views are synthesized by iterating over the triangular facets in the terrain model, projecting the vertices of each triangle to the source and view images, and "warping" each triangular patch in the source image into its corresponding patch in the view image.

The warp step iterates over pixels on the regular grid of the synthesized view that are within each triangle, computing the position of the corresponding pixels in the source image using the linear transformation that maps the triangle in the synthesized view into the source image.

The "warp" operation starts by determining the sampling relationship between pixels in the synthesized view and pixels in the source image. For each triangle in the synthesized view, a circle of one pixel diameter is constructed at any point in the triangle. Since all of the triangles are planar, and the following projections do not include perspective scale change, the particular choice of point does not matter. A cone is constructed by projecting this circle through the projection center of the synthetic camera. This cone is intersected with the corresponding triangular facet of the terrain model, forming an ellipse. A second cone is constructed by projecting this ellipse through the projection center of the camera for the source view. The intersection of this second cone with the image plane of the source view results in an ellipse that corresponds to the circular pixel in the synthetic view (see Figure 1).

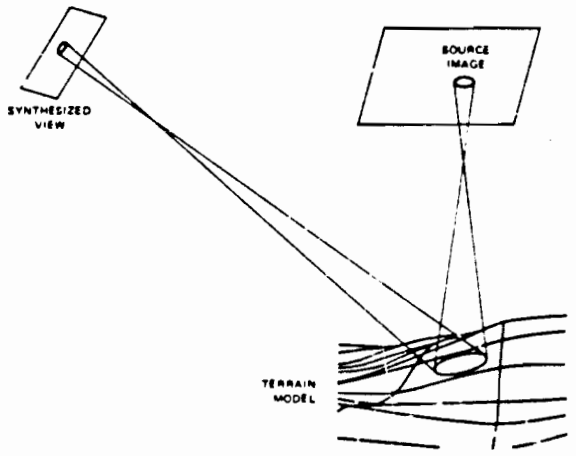


Figure 1: Geometry of the View Synthesis Algorithm

A somewhat more accurate, but also more complicated calculation of the sampling relationship projects a one-pixel-square area from the synthesized view, rather than a circle, and results in a quadrilateral area in the source image.

The use of this sampling relationship is essential to avoid problems due to aliasing, which result from violating the sampling theorem. To avoid aliasing, each pixel in the synthetic view is computed by integrating pixel values in the source image over an elliptical area corresponding to the pixel.

Terrain-Calc computes an approximation to the integral over an elliptical area by summing estimates of integrals over circles that have diameters approximately equal to the minor axis of the ellipse along a path corresponding to the major axis of the ellipse. The circular integrals of various diameters are formed by convolving the source image with circularly symmetric Gaussian convolution kernels of varying sizes using the hierarchical Burt algorithm (Burt, 1981).

Another form of aliasing can occur at pixels that cross occlusion edges, where pixels in the synthetic view project to several facets in the terrain model. The most severe problem of this kind occurs at occlusion boundaries, where the pixel in the synthesized view projects to widely separated facets in the terrain model. The correct calculation requires summing the intensity integrals over two or more partial ellipses corresponding to the intersections of the cone with each facet.

Hidden-surface elimination is accomplished using a variation of the "H array" technique of Wright (Wright, 1973), which requires that (1) the facets be processed in a near-to-far order in relation to the synthetic camera, (2) that the world z-axis always project to a vertical line in the synthetic view (i.e. no roll to the camera), and (3) the geometric model be a single-valued function z(x,y). An improved algorithm (Anderson, 1982), which permits camera roll and fixes some other problems with the H-array technique, will be implemented in the near future. A more conventional z-buffer algorithm would eliminate all of the above restrictions at additional computational cost.

The time required by the view-synthesis algorithm is mainly determined by the number of pixels in the generated view and the number of facets that must be examined. For views containing approximately 320 x 250 pixels resulting from 41000 facets the view-synthesis algorithm requires about 150 seconds on a Symbolics 3600.

INTERACTIVE USER INTERFACE

Terrain-Calc also provides a sophisticated graphical interface for specifying flight paths and parameters of a simulated camera (see Figure 2).

To specify a flight path, the user first invokes an interactive curve editor to draw a curve on top of a vertical view of the terrain model as depicted on the display screen, thereby specifying the x and y components of the flight path. Terrain-Calc then displays a graph of the terrain model profile underneath the flight path, allowing the user to specify the z component of the flight path in relation to the terrain profile. A parametric

Figure 2: (a) Terrain-Calc Showing a Synthesized Image (upper window), the Source DTM with Elevations Displayed by Brightness (lower left window), and the Elevation Profile in the View Direction (lower right window)

Figure 2: (b) An Image Synthesized by Terrain-Calc

spline curve is fit to the x, y, and z components of the flight path, from which position and direction can be easily computed as a function of distance along the curve.

Each synthetic view is computed by means of a perspective projection whose parameters are determined by the flight path and a parameter menu consisting of following.

- *Field of View:* Horizontal field of view. This is the angle relating the focal length to the view image width.
- *Tilt:* Tilt or pitch of the camera in a vertical plane with respect to the direction of the flight path. Currently, tilt must not be large enough to cause any rays from the camera to be exactly vertical, because of limitations of the hidden-surface algorithm.
- *Pan:* Pan or yaw of the camera with respect to the flight path.
- *Sequence Length:* Number of equally spaced views to be generated.
- *Draw Mode:* Selection of wire frame or synthetic image views.

A sequence of views spaced at equal distances along the flight path is generated. For each view, the combination of flight-path direction, tilt, and pan determines the direction of the principal camera ray, which, together with the flight-path position and focal length, determines all of the parameters of the perspective projection for the view. Sequences of views that fit in available physical memory can be dynamically displayed on the color screen at a rate of about 1.3 million pixels per second, or sixteen 320 x 250-pixel frames per second. On a Symbolics 3600 with six megabytes of physical memory, there is room for about 60 frames, each containing 320 x 250 pixels.

Stereo views are created using two identical synthetic cameras separated by a user-specified distance on a horizontal line perpendicular to the direction of the principal ray. They are displayed either as left-right pairs of images for viewing using a stereo viewing box to merge the images or as a cyan/red anaglyph image. Left-right stereo-pair sequences can be displayed at half the above frame rate, whereas anaglyph stereo sequences can be displayed at the full frame rate.

UNSOLVED PROBLEMS AND FUTURE DIRECTIONS

A major unsolved problem is how to improve the efficiency of the projection algorithm by using hierarchical terrain models, in which the level of the hierarchy (and therefore the size of the facets) is chosen in each neighborhood of the terrain model so that there are no noticeable flaws in the generated views. The use of a hierarchy is particularly important for the synthesis of oblique views, where distant facets of the terrain model project to a small fraction of a pixel in the view.

A simple hierarchical technique is to represent the terrain at a hierarchy of resolutions, obtained by convolving the terrain model z(x,y) with Gaussian kernels of various sizes and then decimating the results. The view-synthesis algorithm begins using the highest resolution in the pyramid and, for each row of facets, keeps track of the distance to the nearest facet in the row. As this distance increases, the coarser levels of resolution in the terrain hierarchy are used, in order to keep the number of view-image pixels per facet approximately constant.

This technique produces acceptable results when images are viewed in isolation, but introduces annoying artifacts in motion sequences. The problem is that the transitions between levels of the terrain hierarchy occur at different places in each image, depending on the distances between the facets and the camera. Such transitions occur in jumps. We have implemented an improvement that performs linear interpolation between levels in the resolution hierarchy to eliminate the abrupt transitions.

Currently, Terrain-Cale only handles the very restricted class of geometric models of the form z(x,y). A future extension will allow a mixture of 3-D modeling techniques to be used together, using a Z-buffer (Catmull 1974) or A-buffer (Carpenter) to merge the results of the disparate modeling systems.

REFERENCES

Anderson, David, "Hidden Line Elimination in Projected Grid Surfaces," ACM Transactions on Computer Graphics, October 1982, pp. 275-288.

Blinn, James, "Simulation of Wrinkled Surfaces," SIGGRAPH Proceedings, August 1978, pp. 286-292.

Burt, Peter, "Fast Filter Algorithms for Image Processing," Computer Graphics and Image Processing, May 1981, pp. 20-51.

Carpenter, Loren, "The A-Buffer, an Antialiased Hidden Surface Method," ACM Computer Graphics, July 1984, pp. 103-108.

Catmull, Edwin, "A Subdivision Algorithm for Computer Display of Curved Surfaces," University of Utah, Salt Lake City, December 1974.

Catmull, Edwin and Alvy Ray Smith, "3-D Transformations of Images in Scanline Order," ACM SIGGRAPH'80 Conference Proceedings, July 1980, pp. 279-285.

Quam, Lynn, "Computer Comparison of Pictures," Stanford Artifical Intelligence Project Memo No. 144, May 1971.

Wright, Thomas, "A Two-Space Solution to the Hidden Line Problem for Plotting Functions of Two Variables," IEEE Transactions on Computers, January 1973, pp. 28-33.

GC970145-37

Converting Feature values to Evidence

George Reynolds, Deborah Strahman, Nancy Lehrer

Computer and Information Science Department

University of Massachusetts at Amherst

Abstract

In this paper we describe the mathematical foundations of a knowledge representation framework within the domain of vision using the theory of evidential reasoning as developed by Dempster and Shafer. We describe an evidential form of reasoning and combination rule which is shown to be equivalent to Dempster's rule but linear with respect to the number of elements of the frame of discernment. This is a tremendous computational advantage over the general theory which provides a decision theory exponential with respect to the number of elements in the frame. A preliminary experiment in image interpretation is presented to illustrate the use of the theory.

1. Introduction

In this paper we describe the mathematical foundations of a knowledge representation framework within the domain of vision using the theory of evidential reasoning as developed by Dempster [1968] and Shafer [1976]. This paper is only a summary. No proofs are given and the examples are only outlined. Complete details will be given in a forth coming technical report.

The representation has two components. The first part is static, and explicitly associates measurable properties (eg. features) of the image data, via knowledge sources, to labels which are to be assigned to abstractions of the image data. The second part uses this static representation, a frame of discernment, and the theory of evidence as developed by Shafer to combine the results from the knowledge sources and arrive at a consensus opinion for the purpose of determining the correct label of the image abstraction.

The association between measurements and potential labels of an image abstraction is made using the notion of a mass function as defined by Shafer [1976]. In this paper mass functions generated using explicit knowledge about the image domain in question using possibility functions (see section 6). Methods for making such an association have been made before (Lowrance [1982], Wesley and Hanson [1985], Strat [1984]). However these methods require that the range of values over which the mass functions are defined be either explicitly or implicitly discretized into "feature propositions" or subintervals of the feature variable.

In our approach no such discretization is required and the process of creating a mass function is defined as a continuous variable of a feature value. Moreover we will define a combination process and a choice of decision rules which are linear with respect to the number of elements of the frame of discernment. This is a tremendous computational advantage over the general theory which provides a decision theory exponential with respect to the number of elements in the frame.

Our approach is only one example of how mass functions might be generated from measures on features. It is doubtful that there is a domain independent way of creating mass functions from feature measurements, since the domain imposes constraints on the semantics of the desired inference process, and some ways of generating mass functions may not satisfy these constraints.

The formalism we develop in this paper is not a strictly probabilistic approach but uses a product rule to combine information from knowledge sources and in this sense can be viewed as a "generalized Bayesian" method. A recent paper of Hummel [1985] clarifies the relationship of Dempster's rule to more traditional "Bayesian" methods in terms of sets of "experts". The correct use of the Dempster-Shafer formalism involves carefully stating the assumptions about the domain to which it applies and representing these assumptions within the knowledge network via possibility functions. If this representation is faithful to the assumptions, then Dempster's rule will focus mass on consistent statements and thus "preserve consistency". An important aspect of the representational formalism involves the use of a "conflict value" which detects when an assumption has been violated and is used as representation of uncertainty within the system. The system also provides a simple mechanism to isolate groups of propositions which are mutually consistent.

The combination process can also be viewed as generating a new higher level "combined" feature, such as might be obtained combining intensity, texture, color and edge features. This is a very useful attribute of the system since it separates the combination process from some other process or processes which would be responsible for executing a decision procedure in order to determine which label is correct on the basis of this new combined feature.

The effect of applying Dempster's rule to mass functions which involve different partially restricted subsets is one of focusing that mass or belief onto the consistent subsets of possibilities. Used this way, the ideas of Shafer and Dempster become a process of possibilistic reasoning rather than probabilistic reasoning. The mass attributed to impossible situations is viewed as conflict or mass assigned to an "unknown" object; it may be monitored to evaluate the correctness of the knowledge sources, and the knowledge base (frame of discernment).

2. Knowledge Representation using Frames of Discernment

Suppose we are presented with a question and a finite set, Θ, consisting of possible answers to the question, only one of which is the correct one. Then for each $\theta \in \Theta$ the proposition of interest is precisely of the form "*The correct answer is* θ". A set will be called a *frame of discernment* (to be defined exactly below) when its elements are interpreted as possible answers to a particular question, and we know that exactly one of the answers is correct. Each subset $P \subseteq \Theta$ can be interpreted as a proposition which states : "*The correct answer is in the set* P". Thus the set of all propositions relevant to finding the correct answer is in a one to one correspondence with the set of subsets of Θ, i.e. 2^Θ.

Now suppose that, in addition, we can make measurements on the environment to which the question pertains such that the outcome of those measurements allows us to rule out some of the elements of Θ as being the correct answer. Then if the outcome of measurement M_1 implies that the correct answer lies in $P_1 \subseteq \Theta$ and measurement M_2 implies that the correct answer lies in $P_2 \subseteq \Theta$ then the correct answer will lie in $P_1 \cap P_2$. Thus the problem of finding the answer to the question becomes one of finding the features of the environment in question which can be measured and determining which of those can give us partial information about what is the correct answer. The correct answer can then be found if enough features can be identified that allow a singleton of the set Θ to be obtained when the corresponding intersection of the P's is computed.

In the context of image understanding for example, the question might be: "What is the correct label to assign to a region of an image?". In the context of outdoor scenes we might have $\Theta = \{foliage, grass, roof, shutter, sky\}$ and the problem is to find features which can assist in finding the correct label. Typically the information that is relevant to answering this question is contained in regions, lines and other image abstractions that are produced from an analysis of the raw image (or other sensor data) by the low-level system. In particular there are attributes or features of these intermediate level structures such as size, texture, shape, length of adjacent lines, region adjacency conditions and spatial relations which contain the constraints needed to identify the correct label.

The problem with the simple model described above is that although it allows a knowledge source to assign its evidence to a subset of Θ (as opposed to a probability distribution on Θ) it contains no representation of uncertainty. The representational formalism we are developing is to provide a way of connecting measurements which we can make in an environment and a question we are trying to answer about that environment in a way which includes a representation of any uncertainty in the measurement or the implications of the measurement. Measurements are converted into mass functions and these functions combined using Dempster's rule. Let us briefly review the definitions involved.

Definition: A mass function is a function

$$M: 2^\Theta \rightarrow [0,1]$$

so that

$$M(\phi) = 0$$

$$\text{and} \quad \sum_{A \subseteq \Theta} M(A) = 1$$

Recalling that set Θ is to be a frame of discernment when its elements are interpreted as possible answers to a particular question, and we know that exactly one of the answers is correct, then for a given set $P \subseteq \Theta$, $M(P)$ should be interpreted as the amount of belief or evidence M has that P is the set every element of which the evidence supports as being the the correct answer. Below we will see that each measurement we make on an environment will generate a mass function which is the evidence that that measurement provides as to which sets contain the correct answer. Thus measurements of two different features will provide two different mass functions each of which is the distribution of a unit of belief for which sets contain the correct answer. Dempster's rule is a way of combining mass functions, the result being another mass function which focuses the mass on the set which both measurements support as the set containing the possibly correct answers.

Dempster's Rule: If M_1 and M_2 are mass functions then

$$M_1 \oplus M_2 (C) = \frac{\sum_{A \cap B = C} M_1(A) \cdot M_2(B)}{1 - k}$$

where

$$k = \sum_{A \cap B = \phi} M_1(A) \cdot M_2(B).$$

$M_1 \oplus M_2$ is called the **combination** of M_1 and M_2, k is called the **conflict value**, and the combination is defined if and only if $k \neq 1$. If we interpret the two mass functions as each distributing a unit of belief for the largest sets which a knowledge source believes contain the correct answer, then $k = 1$ if and only if the evidence provided by M_1 flatly contradicts the evidence provided by M_2. In general the conflict value is a measure of the extent to which two bodies of evidence contradict each other with $k = 0$ precisely when they are consistent.

A frame of discernment and mass functions, together with Dempster's rule, provide a way of representing two kinds of uncertainty. First, the frame of discernment allows each feature measurement to be associated with a subset $P \subseteq \Theta$ containing the set of answers which are possibly correct without committing any belief to the elements of P prematurely, (Thus disbelief is distinguished from no belief). Second, mass functions allow for the evidence to be distributed to more than one subset and thus individual measurements can suspend the impact of their uncertain outcomes until the evidence is pooled using Dempster's rule.

The conflict value k may be non-zero for one of two reasons. On the one hand the evidence represented by the mass function may be in error, or on the other, one of the assumptions implied by the representation of knowledge within the frame of discernment has been violated. In the latter case it may be that the frame of discernment and the process of generating the mass

functions needs to be modified to correctly reflect the assumptions of the domain, or that an event has occurred which was not included in the frame and Θ needs to be enlarged.

If a set $A \subseteq O$ is assigned mass t, then any set B with $A \subseteq B \subseteq \Theta$ should believe an amount at least t that it too contains the right answer. In addition any set $C \subseteq \Theta$ with $C \cap A = \phi$ should have the extent to which the evidence refutes C as containing the right answer reduced by at least t. This leads to the following definition:

Definition Given a mass function $M: 2^\Theta \rightarrow [0,1]$, the **support** and **plausibility** of each $A \in 2^\Theta$ is defined as follows:

$$SPT(A) = \sum_{X \subseteq A} M(X)$$

$$PLS(A) = 1 - \sum_{X \cap A = \phi} M(X).$$

3. The structure of an evidential knowledge base

Let us review some of the limitations of inferencing using Bayesian probability models, especially in the domain of vision. Central to the problem of image understanding is how to convert a feature value into evidence for an object or set of objects given knowledge about the relationship of a feature to the set of objects. For example Bayes rule is one way of making such a conversion: Suppose we are given a set of labels $a, b, c, \ldots$ and probabilities $p(a), p(b), p(c), \ldots$ and for feature f, $p(f \mid a), p(f \mid b), p(f \mid c), \ldots$ and $p(f)$. Then

$$p(a \mid f) = \frac{p(f \mid a)p(a)}{p(f)}.$$

This formula presents a number of problems. Consider the problem of assigning labels to the regions of some image segmentation. If for example the labels are *partof-sky*, *partof-road*, *partof-grass*,... then *p(partof-sky)* would be the probability that a region in the segmentation is correctly labeled *partof-sky*. Thus not only is it necessary to have knowledge about the frequency of the event sky, it is also necessary to have knowledge about the whims of the specific segmentation algorithm being used and its effect on the prior probability *p(partof-sky)*. Experience with segmentation algorithms suggests that this information is extremely difficult to obtain.

On the other hand, it is possible to obtain some knowledge about $p(f \mid a)$ although the statistics used to estimate this number are subject once again to the variations imposed by the segmentation algorithm being used, variables which are very difficult to characterize statistically. Thus although there is some useful information in these statistics there is a tremendous amount of uncertainty in the histograms. In addition it is possible to acquire estimates of $p(f)$ and the Rule System of Hanson, Riseman, et al. [1985] uses the ratio

$$\frac{p(f \mid a)}{p(f)}$$

and approximations of this ratio as a "vote" for a when the feature f is observed. However the rule system is limited in its ability to handle uncertainty, partial ignorance and conflict between knowledge sources.

Consider the problem of labeling an image with labels $a \in A$. The distribution $p(f \mid a)$ provides some information about how often a feature f occurs with respect to the object a. If the frequency is high, for some value f, then at least we don't want to rule out the possibility that the correct label to be assigned is a. On the other hand the feature value f may occur frequently for many objects and so the only knowledge we may have is of the form: *Given an observation f, then we don't want to rule out the possibility that the correct label is in the set A of labels for which that feature occurs frequently.* Combining the information from many such knowledge sources then leads to the correct interpretation.

Definition Given a frame of discernment Θ and a feature space FS, a possibility function

$$p_s(f \mid a) : FS \rightarrow [0,1]$$

is a function defined on a feature space FS for each $a \in \Theta$ which has the interpretation: $p_s(f \mid a)$ is the extent to which we don't want to rule out a if we make the observation $f \in FS$.

Example Figure 3.1 contains an example of a segmentation used to generate the histograms in figure 3.2, 3.3 and 3.4. Regions in 3 segmentations of similar images were selected and identified as foliage, grass, sky, shutter and roof. The features of raw-blue mean, short line density, edge density, centroid row position and excess green mean were then histogrammed. In the figures, the possibility functions are overlaid on these histograms and they are presented in the order foliage, grass, sky, shutter and roof.

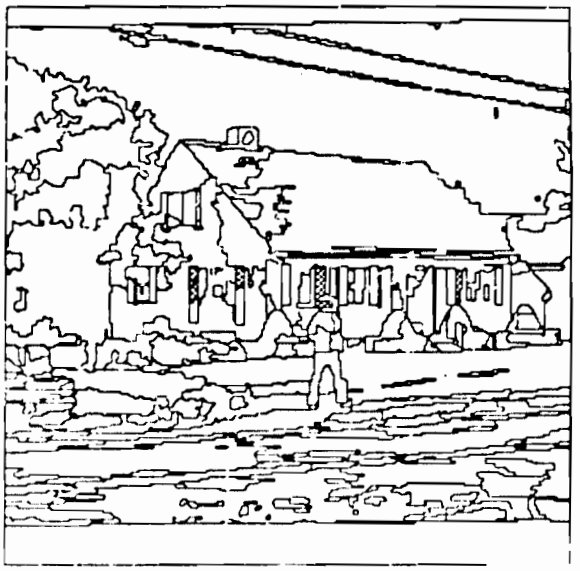


Figure 3.1: Image Segmentation

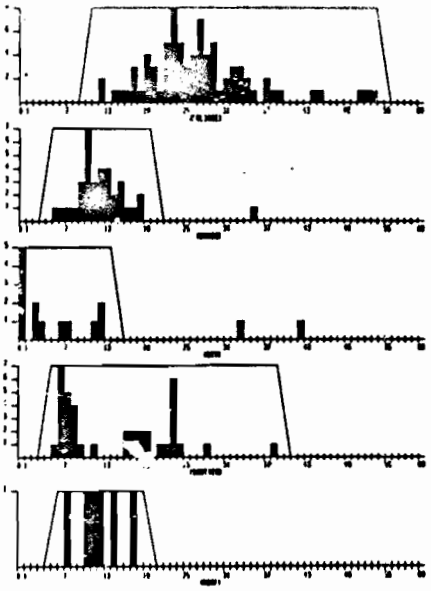


Figure 3.2: Histogram and possibility function for edge density

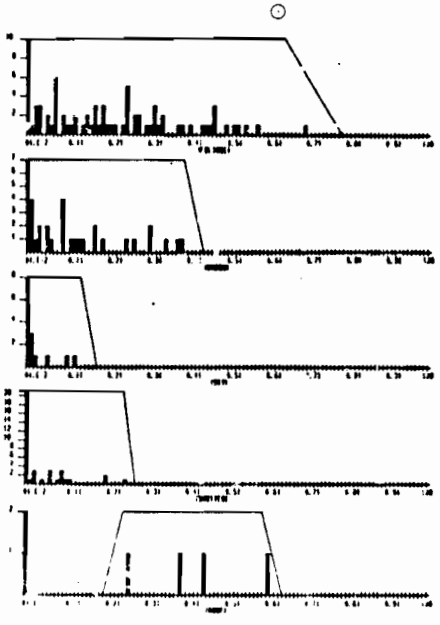


Figure 3.3: Histogram and possibility function for short line density

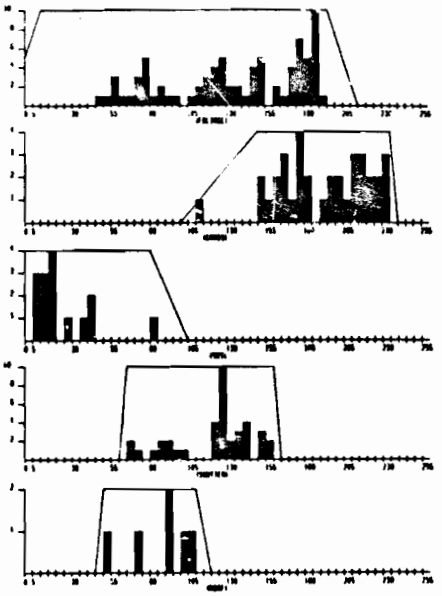


Figure 3.4: Histogram and possibility function for centroid row position

Examining this example we can identify a number of sources of uncertainty. First of all the statistical information is inherently incomplete. It seems clear that there is a much wider variation in the data, even in the narrow context in which these images are a sample. Thus the specification of the possibility function will involve some uncertainty. Second the set Θ is also inherently incomplete. We simply may not have anticipated all the objects we will encounter in a specific context. These uncertainties translate into a requirement that the system to be able to say *"I don't know"* in at least two important ways.

First, an individual knowledge source may be uncertain about the applicability or relevance of a measurement both with respect to the information in the knowledge base and with respect to its own internal processes. In other words an individual knowledge source needs the ability to say *"I don't know"*. (This is separate from the fact that the value supplied by the knowledge source might be in error.)

Secondly, even if each individual knowledge source is completely certain about how its information should be interpreted, there may be conflict and inconsistency between the knowledge sources. In other words we need the ability for the knowledge sources to collectively say: *"I don't know"*. This type of inconsistency can come in a number of forms. For example it may be that 20 knowledge sources are all supplying information at a given time and one of the knowledge sources being in conflict with the other 19 is not to be interpreted as completely invalidating the outcome of the combination process. We need a measure of the degree to which the knowledge sources are collectively uncertain about the consensus opinion. What action to be taken as a function of this uncertainty is yet another matter.

4. Converting Measurements into Mass Functions

We are now in a position to define precisely what we mean by a frame of discernment. A frame is designed to capture the relationships between the objects in some context and the features in that context which pertain to reasoning about those objects. As the context changes, the objects, the features and the relationships between the features and the objects can be expected to change.

Definition: A knowledge source is a function

$$ks : FS \rightarrow M(2^{\Theta})$$

where $M(2^{\Theta})$ is the set of all mass functions on Θ.

Definition: A frame of discernment (or **a context**) is a specification of a set Θ and a collection of knowledge sources

$$ks_1 : FS_1 \rightarrow M(2^{\Theta}), \ldots, ks_n : FS_n \rightarrow M(2^{\Theta}).$$

Each feature space can be thought of as containing quantities that are associated with some observable and quantifiable aspect of the knowledge we are bringing to bear on the problem of answering the question which the frame is designed to answer. The set of all feature spaces of potential interest and their forms a frame of discernment. In general this includes any aspect of a domain or world about which information may be obtained in order to help decide which answer is correct. In our approach to reasoning about one's environment, various types of knowledge sources provide the partially processed information, based on their environmental observations, about the "evidence for" or "belief in" the propositions represented by the subsets of Θ.

Suppose we are given a set Θ and for each $a \in \Theta$ a possibility function. We will now give a construction which generates a knowledge source form this possibility function. In fact there are a number of ways to modify this construction which also yield mass functions which may in fact turn out to be very useful in some circumstances. Thus the construction we give here is a starting point for a whole class of constructions. The advantage of the method we describe is that it allows the computation of the supports and plausibilities for the elements of Θ without the need of the power set. Moreover we will describe a combination function which yields the same result as Dempster's rule but is linear with respect to the number of elements of Θ. This will lead to a decision rule based on the supports and plausibilities whose complexity is linear in the number of elements of Θ.

Definition Suppose we are given a set Θ and for each $a \in \Theta$ a possibility function

$$ps(f \mid a) : FS \rightarrow [0,1],$$

from some feature space FS to $[0,1]$. Then for each $A \subseteq \Theta$ define

$$m_0(A \mid f) = \prod_{a \in A} ps(f \mid a) \prod_{a \in \Theta - A} (1 - ps(f \mid a)).$$

Now the function $m_0(A \mid f)$ is **not** a mass function since

$$m_0(\phi \mid f) = \prod_{a \in \Theta} (1 - ps(f \mid a))$$

is not necessarily equal to zero. However the function does have the other two properties of a mass function:

1. $0 \leq m_0(A \mid f) \leq 1$,
2. $\sum_{A \subseteq \Theta} m_0(A \mid f) = 1$.

The first statement is obvious and the following lemma directly implies the second.

Lemma 4.1 *If $(z_1, \ldots, z_n)$ is a sequence of numbers and $N = \{1, \ldots, n\}$ then*

$$\sum_{A \subseteq N} \prod_{i \in A} z_i \prod_{j \in N - A} (1 - z_j) = 1$$

where we define

$$\prod_{\phi} z_i = 1$$

Consider what it means for the empty set to receive a non-zero value in terms of the possibility functions generating m_0. It means simply that the consensus of opinion of the possibility functions is that the feature value in question rules out every element of Θ with respect to the current state of the knowledge base (as represented by the possibility functions). This could be either because the knowledge source is in error or the knowledge base is incomplete. The decision as to which of these conditions holds is external to the processes of the inference network. All that should be required of it is that it return (partially) the answer *unknown*.

Therefore we add to Θ a new element *unk* and define

$$m(A \cup \{unk\} \mid f) = m_0(A \mid f).$$

This then, is a mass function on $\Theta \cup \{unk\}$

We could have recast our definition by defining a new object (a pre-mass function?) which is allowed to assign non-zero mass to the empty set (see Hummel [1985]). Dempster's rule can be defined for these objects (just take out the re-normalization) and there is a simple mapping between these objects and mass functions. However, this approach requires doubling the notation. The addition of $\{unk\}$ requires no change in notation or conceptualisation, eliminates the need to re-normalize until it is appropriate and the "conflict" value generated by Dempster's rule is simply the mass assigned to $\{unk\}$.

The following theorem summarizes some of the relationships between the possibility values and the generated mass function.

Theorem 4.1 *Suppose we are given a set Θ and for each $a \in \Theta$ a possibility function*

$$ps(f \mid a) : FS \rightarrow [0,1],$$

from some feature space FS to $[0,1]$. Then defining the mass function $m(A \cup \{unk\} \mid f)$ as above,

1. $spt(A) = 0$ *if* $unk \notin A$,
2. $pls(\{a\}) = ps(f \mid a)$ *for any* $a \in \Theta$,
3. $spt(\{a, unk\}) = \prod_{x \in \Theta}(1 - ps(f \mid x)) + ps(f \mid a)\prod_{x \neq a}(1 - ps(f \mid x))$ *for any* $a \in \Theta$,
4. $pls(A \cup \{unk\}) = 1$ *for any* $A \subseteq \Theta$.

Given a mass function as defined above, we can define a function on 2^Θ by the formula

$$m - norm(A \mid f) = \frac{m(A \cup \{unk\} \mid f)}{1 - m(\{unk\} \mid f)}$$

for any non-empty set A and $m - norm(\phi \mid f) = 0$.

Theorem 4.2 *$m - norm$ is a mass function on 2^Θ.*

5. Possibility functions and Dempster's rule

In this section we make the connection between possibility functions and Dempster's rule. In particular we will show that combining possibility functions by term-wise product and generating a mass function yields the same result as individually generating mass functions and combining using Dempster's rule.

Let us briefly consider some results which are useful for deriving computationally efficient ways of computing Dempster's rule, supports and plausibilities. The first observation is that if we are presented with a set of mass functions on $\Theta \cup \{unk\}$ to combine, and they only assign non-zero mass to subsets of $\Theta \cup \{unk\}$ which contain unk, then the amount of mass which accumulates on $\{unk\}$ is exactly the conflict value of the n-wise combination as defined by Shafer.

Next we observe that normalization and combination commute with each other.

Theorem 5.1 *Suppose $m_1(A \cup \{unk\} \mid f)$ and $m_2(A \cup \{unk\} \mid f)$ are mass functions derived from the possibilsty functions $ps_1(f \mid a)_1$ and $ps_2(f \mid a)$. Then*

$$(m_1 \oplus m_2) - norm = m_1 - norm \oplus m_2 - norm$$

Theorem 5.2 *Suppose $m(A \cup \{unk\})$ is a mass functions derived from the possibility function ps. Then with respect to the mass function $m - norm$,*

$$pls(\{a\}) = \frac{ps(f \mid a)}{1 - \prod_{a \in \Theta}(1 - ps(f \mid a))},$$

$$spt(\{a\}) = \frac{ps(f \mid a) \prod_{b \in (\Theta - \{a\})}(1 - ps(f \mid b))}{1 - \prod_{a \in \Theta}(1 - ps(f \mid a))}.$$

Thus we have shown the connection between possibility functions and mass functions. The last theorem in particular shows that with respect to the elements of Θ the supports and plausibilities can be computed directly from the possibility functions without the need of the power set. Thus any decision rule based on the support and plausibility has a complexity which is proportional to the number of elements of Θ (see Wesley and Hanson [1985]).

The next two theorems show that term-wise product of possibility functions is equivalent to combining using Dempster's rule. We first observe that every mass function generated by a possibility function is a separable mass function.

Theorem 5.3 *Suppose we are given a set Θ and for each $a \in \Theta$ a possibility function*

$$ps(f \mid a) : FS \rightarrow [0, 1],$$

from some feature space FS to $[0, 1]$. Then the mass function $m - norm$ generated by this possibility function is a separable mass function.

Theorem 5.4 *Suppose we are given a set Θ and for each $a \in \Theta$ possibility functions*

$$ps_1(f \mid a) : FS \rightarrow [0, 1],$$

and

$$ps_2(f \mid a) : FS \rightarrow [0, 1],$$

from some feature space FS to $[0, 1]$. Let

$$ps_3(f \mid a) = ps_1(f \mid a) \cdot ps_2(f \mid a)$$

Then defining the mass functions m_1, m_2 and m_3 in terms of these possibility functions as above,

$$m_3 = m_1 \oplus m_2.$$

6. A preliminary experiment

We now present a discussion of some preliminary results of interpreting outdoor house scenes. In this experiment, the question asked by our system is *Which regions of this segmentation represent foliage, grass, sky, shutter, and roof?* To answer this question a frame of discernment was developed heuristically based on the histograms of several feature spaces for hand picked regions from similar segmentations.

First we will outline the decision rule which was used to interpret the image. Suppose we are given a mass function M generated (say) by an evidence combination process described in the previous sections. In this paper we will not develop a decision theory based on the SPT and PLS, however we will state the simple criterion which was used in the interpretation experiment.

Definition Given a mass function $M: 2^\Theta \to [0,1]$, the **decisiveness** of each $A \in 2^\Theta$ is defined as

$$DEC(A) = SPT(A) - (1 - PLS(A)).$$

Note that $DEC(A)$ is a number between -1 and 1. If the $SPT(A)$ is close to 1 then $DEC(A)$ is close to 1 and of the $PLS(A)$ is close to 0 then $DEC(A)$ is close to -1, and in particular if $SPT(A) = 0$ and $PLS(A) = 1$ then $DEC(A) = 0$. Thus if $DEC(A)$ is close to 1 then the evidence supports A, if $DEC(A)$ is close to -1, the evidence tends to refute A, and if $DEC(A)$ is close to 0 then the evidence is indecisive. One simple decision criterion then is to compute the decisiveness for each singleton of Θ and take the element which has the maximum value That is select $a \in \Theta$ where

$$DEC(\{a\}) = max\{DEC(\{x\}) \mid x \in \Theta\}$$

This measure preserves the ordering implied by the possibility values, and can be used even when the possibility values are unavailable.

Now consider the situation where for a given feature value to be interpreted by a particular knowledge source, the mass assigned to the unknown object is not equal to zero. When this happens we say the knowledge has internal conflict and this value can be viewed as the "uncertainty" of the knowledge source. This arises when every object is given a possibility value less than 1 by the knowledge source. For example given the possibility functions in figure 5.2 for edge-density, if the feature value returned is above 55, then the knowledge source will in effect say, *"For this region there is no object defined."* and assign 1.0 mass to the singleton *(unknown)*. The amount of internal conflict can be calculated directly from the possibility function by the following formula.

$$k = m_\theta(\phi \mid f) = \prod_{a \in \Theta} (1 - pos(f \mid a))$$

While combining the evidence provided from all knowledge sources the uncertainty of a single knowledge source will be also reflected in the combined conflict. Intuitively, it seems reasonable that the "conflict" created by an individual knowledge source should however be distinguished from the conflict created by the consensus of knowledge sources.

One simple method for dealing with undefined knowledge is to simply remove any knowledge source with a significantly large internal conflict. Unfortunately this presents the problem of determining an appropriate threshold for the internal conflict and all information contained in that knowledge source is lost by this process when the conflict is above that threshold.

One approach to the problem stated above is to condition each possibility function by the degree to which its knowledge is defined before the combination process. Such a conditioning is obtained by using the plausibility of the mass function $m-norm$:

$$pls(\{a\}) = \frac{pos(f \mid a)}{1 - k}$$

With k as defined above (see Theorem 6.1).

The "weighting" process allows a possibility function with total uncertainty (for example $(\langle a\ 0.0\rangle, \langle b\ 0.0\rangle)$) to have no effect on the consensus of opinion, and a possibility function with no uncertainty to have full effect on the consensus. The amount of weight given to a possibility function is a continuous function of the internal conflict in the knowledge source. Although this approach deals partially with the problem discussed above, a complete decision rule must be a function of the internal conflicts of the knowledge sources, the combined conflict, the support and plausibility.

For the experiment presented here, all knowledge sources are assumed to be independent and we are most interested in a decision based on the conflict created between knowledge sources. Thus, in this case we have chosen to base the decision rule on the combined conflict and the decisiveness, diminishing the effect of the internal conflict in the manner described above. This leads to the following decision used for interpreting each region of the image:

1. Run each knowledge source for a region and return a list of possibility values defined over Θ.
2. Re-normalise the possibility lists by the conflict internal to each knowledge source.
3. Combine the possibility lists (via the multiplicative parallel to Dempster's Rule) and convert the result into a mass function defined over 2^Θ.
4. Re-normalise the resulting mass function by the combined conflict.
5. Label each region based on the combined conflict measure and the decisiveness values defined over the singleton subsets of Θ.

If the combined conflict was above some threshold t (in this case 0.25) then the final labeling for that region is *unknown*. Otherwise the final label was chosen by computing the decisiveness values for each singleton. The singleton with largest decisiveness value or the subset containing the two or more singletons with the same largest decisiveness value was chosen as the label for the region under discrimination. The results are shown in figures 6.1 thru 6.9 where the regions are labeled *foliage, grass, foliage or grass, shutter, foliage or shutter), roof, foliage or roof, sky,* and *unknown*.

7. Conclusions

In this paper we have considered mass functions generated from possibility functions defined from the a priori statistics of features and objects. It is obvious that the mass assignments generated from possibility functions bear a great resemblance to probabilities on sets of independent events. For the examples given, this form is intuitively appealing as well as compact and easily analyzed. However, not all relationships between image features and their interpretations can be captured by the use of possibility functions in the way that we have defined the relationship between possibility functions and mass functions. We will conclude by considering two situations where this occurs.

First, a coarsening or refinement of the frame of discernment may be required. In this case the mass function on the refinement can not necessarily be generated by a possibility function over the refinement. For example, in the context of aerial photographs, a measure of rectangularity may discern between rectangular and non-rectangular objects but it is not appropriate to use this measure to distinguish between potentially rectangular objects (such as buildings or parking-lots). Therefore a single frame of discernment and related knowledge sources can not be used throughout the reasoning process and the system must be able manange mass functions of broader types than mentioned here.

Shafer [1976] suggests that in situations where the combination of mass functions produces a great deal of conflict the individual mass functions can be discounted then recombined. An example of this is uniform discounting which reduces the mass given to each proper subset and increases the mass given to Θ. If the discounting factor and conflict is large enough then the combined mass to each proper subset tends toward an "average" of individual mass functions. The discounted mass function is not necessarily separable into the simple mass functions of the form described above, and thus the analysis using possibility functions does not apply.

Thus mass functions generated by possibility functions form only a proper subset of the mass functions which are appliciable in a general image understanding system, however their simplicity and computational advantages make them very attactive in contexts where evidential reasoning and management of uncertainty is required.

Acknowledgements

We would like to thank Len Wesley, Al Hanson and Les Kitchen for many invaluble conversations concerning this paper. We would especially like to thank Les for suggesting that we consider the combination rule for possibility functions. The work reported here was supported by the Air Force under AFOSR contract no. F49620-83-c-0099

References

A. P. Dempster (1968), "A generalization of Bayesian inference", *Journal of the Royal Statistical Society*, Series B, vol. 30, 1968, pp. 205-247.

A. Hanson, E. Riseman, J. Griffith, T. E. Weymouth (1984), "A methodology for the development of general knowledge-based vision systems", *Proc. IEEE Workshop on Principles of Knowledge Based Systems*, Denver, Colorado, Dec. 1984, pp. 159-170.

R. A. Hummel (1985), "A viewpoint on the theory of evidence", to appear.

J. D. Lowrance (1982), "Dependency-graph models of evidential support", Ph.D. Thesis, University of Massachusetts, Amherst, 1982.

S. Y. Lu, H. E. Stephanou (1984), "A set-theoretic framework for the processing of uncertain knowledge", AAAI-84, pp. 216-221.

G. Shafer (1976), *A Mathematical Theory of Evidence*, Princeton University Press, 1976.

T. Strat (1984), "Continuous belief functions for evidential reasoning", AAAI-84, pp. 308-313.

L. Wesley (1983), "Reasoning about control: the investigation of an evidential approach", *Proc. Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, August 1983, pp. 203-210.

L. Wesley (1984) "Reasoning about control: an evidential approach", Tech. Report 324, SRI Artificial intelligence Center, 1984.

L. Wesley (1985), Ph.D. Thesis, University of Massachusetts, Amherst, *in preparation*.

L. Wesley, A. Hanson (1985), "The application of an evidential-based technology to a high-level knowledge-based image interpretation system", to appear.

Figure 6.1 Grass

Figure 6.2 Foliage

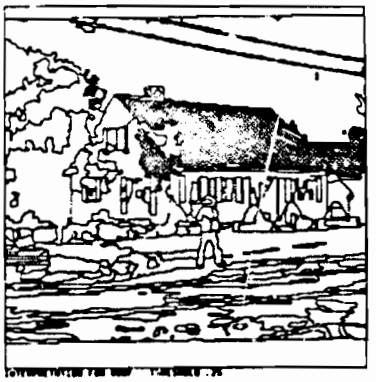


Figure 6.3 Roof

Figure 6.4 Shutter

Figure 6.5 Sky

Figure 6.6 Unknown

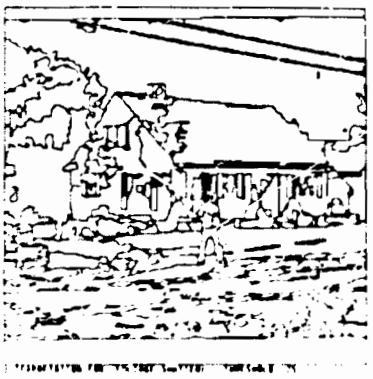


Figure 6.7 Foliage or Shutter

Figure 6.8 Foliage or Grass

Figure 6.9 Foliage or Roof

GC970145-38

BINOCULAR IMAGE FLOWS

Allen M. Waxman*

Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742

and

James H. Duncan

Flow Research Company
Silver Spring, MD 20910

ABSTRACT

The analyses of visual data by stereo and motion modules have typically been treated as separate, parallel processes which both feed a common viewer-centered 2.5-D sketch of the scene. When acting separately, stereo and motion analyses are subject to certain inherent difficulties. Stereo must resolve a combinatorial correspondence problem and is further complicated by the presence of occluding boundaries, while motion analysis involves the solution of nonlinear equations and yields a 3-D interpretation specified up to an undetermined scale factor. A new module is described here which unifies stereo and motion analysis in a manner in which each helps to overcome the other's shortcomings. One important result is a *correlation between relative image flow (i.e., binocular difference flow) and stereo disparity*; it points to the importance of the ratio $\dot{\delta}/\delta$, rate of change of disparity $\dot{\delta}$ to disparity δ, and its possible role in establishing stereo correspondence. The importance of such ratios was first pointed out by Richards (1983). Our formulation may reflect the human perception channel probed by Regan and Beverley (1979).

1. INTRODUCTION

In decomposing the visual information processing task into several stages, it is the intermediate level which is responsible for the recovery of surface shapes in a scene (Marr 1982). It is often described as a set of "shape from" modules which, acting independently and in parallel, feed a viewer centered "2.5-D sketch" of the visual field. Two of the most commonly studied and closely related modules are *shape from stereo* (Koenderink and van Doorn 1976; Marr and Poggio 1979; Mayhew and Frisby 1981; Prazdny 1984; Pollard et al. 1985; Eastman and Waxman 1985) and *shape from monocular motion* (Koenderink and van Doorn 1975; Ullman 1979; Prazdny 1980; Longuet-Higgins and Prazdny 1980; Longuet-Higgins 1981; Tsai and Huang 1981a,b; Waxman and Ullman 1983; Waxman 1984; Waxman and Wohn 1984; Wohn and Waxman 1985a,b; Subbarao and Waxman 1985; Buxton et al., 1984). However, when acting independently, each of these processes suffers from certain inherent difficulties. Stereo is faced with a combinatorial correspondence problem plagued by the presence of occluding boundaries (Grimson 1981; Poggio and Poggio 1984), while motion analysis involves the solution of nonlinear equations and leaves the 3-D interpretation specified up to an arbitrary scale factor (Waxman and Ullman 1983). There is evidence, however, for a separate channel of human visual processing in which stereo and motion analyses may come together much earlier than at the 2.5-D sketch. We formulate here a theory of time-varying stereo in the context of "binocular image flows," where stereo and motion work closely in order to overcome each other's shortcomings. Central to our approach is the notion of *relative flow* (or "binocular difference flow"), representing the difference between image velocities of a feature as seen in the left and right images separately. Neural organizations which perform this "computation" have already been proposed (Regan and Beverley 1979).

The fusion of stereo and motion into a single module has been considered recently by others as well. Richards (1983) demonstrated recovery of structure from orthographic stereo and motion without knowledge of the fixation distance. He also pointed out the importance of measuring changes in quantities, such as disparity over time, relative to their current values. Jenkin (1984) considered a stereo matching process driven by the 3-D interpretation of feature point velocities. Waxman and Sinha (1984) proposed a "dynamic stereo" technique based upon the relative flow derived from two cameras in known relative motion, valid in the limit of negligible disparity. The question of image motion aiding stereo in the matching process was noted by Poggio and Poggio (1984); and as will be shown below, a correlation between binocular difference flow and disparity may support this possibility.

We suggest a decomposition of our stereo-motion module into five steps which begins where low-level vision ends, i.e., it follows the stage of edge and point feature extraction (and tracking over time) in the left and right images separately.

Step 1 Monocular image flow recovery and flow segmentation of the separate left and right image sequences utilizing the *Velocity Functional Method* (Waxman and

* Present address: Thinking Machines Corporation, 245 First Street, Cambridge, MA 02142

Wohn 1984) and *overlap compatibility* (Waxman 1984; Wohn and Waxman 1985b). This procedure allows gross correspondence to be established between analytic flow regions in the left and right images. It also reveals the depth and orientation discontinuities that often plague stereo matching and surface reconstruction algorithms.

Step 2: Establishing correspondence between (previously unmatched) left and right image features according to a correlation between binocular difference flow and stereo disparity. This process can be implemented in parallel over the binocular field of view in the context of "local support" within neighborhoods (Prazdny 1981; Pollard et al. 1985; Eastman and Waxman 1985). This correlation points to the importance of the ratio δ/δ_t, rate of change of disparity δ to disparity δ_t. The importance of this ratio was first noted by Richards (1983). A "rigidity assumption" for independently moving objects in the scene also enters here.

Step 3: Use of disparity functionals defined in overlapping neighborhoods to recover smooth surface structure between the discontinuities detected from the monocular flow analyses (Koenderink and van Doorn 1976; Eastman and Waxman 1985).

Step 4: Recovery of rigid body space motions corresponding to separate analytic flow regions utilizing the determined surface structure and either monocular image flow (or a cyclopean image flow). Separate surface patches can then be grouped into rigid objects sharing the same space motions. This process entails solving only linear equations as a measure of its complexity.

Step 5: Use of the separate image flows to track features and discontinuities over time. This allows refinement of disparity estimates to "sub-pixel" accuracy by temporal interpolation. It also allows the *matching process to focus attention* onto areas where new image features will be unveiled and old ones will disappear, i.e., *at the discontinuities and periphery of the field of view*.

This last step suggests that, in the analysis of a time-varying stereo sequence, once an initial correspondence has been determined between left and right images, it is not necessary to establish correspondence anew for the entire image pair at subsequent times. Most of the image features merely flow to new locations which can be predicted. Matching need only be performed on new features which enter the visible field from the periphery and from behind occluding boundaries.

In this paper we formulate several of these steps toward stereo-motion fusion. Section 2 reviews the basic monocular image flow relations for rigid bodies in motion. The importance of locally second-order flows and boundaries of analyticity (i.e., weak and strong flow discontinuities) is stressed as it is important for the binocular flow analysis that follows. In Section 3 we develop the theory of binocular image flows in the context of a parallel stereo configuration, imaging a scene of rigid objects in motion. A correlation is derived between relative flow (binocular difference flow) and stereo disparity, laying the basis for a new kind of matching procedure. In Section 4 we utilize an experimental data set for a short stereo sequence to obtain the measured binocular image flows at one time instant. These flows are then filtered using the Velocity Functional Method, and a flow segmentation is derived in order to detect depth and orientation discontinuities in the scene. This data is then used to confirm the correlation between binocular flow and disparity developed earlier. We conclude in Section 5 with a discussion of what remains to be done in the construction of a complete stereo-motion fusion module.

2. MONOCULAR IMAGE FLOWS

Investigations into the recovery of 3-D structure and motion from time-varying monocular imagery have proceeded along two rather distinct paths. One approach has been concerned with the motion of discrete points moving rigidly in space (Ullman 1979; Prazdny 1980; Longuet-Higgins 1981; Tsai and Huang 1981a,b; Adiv 1984). The resulting 3-D interpretation is in the form of rigid body motion parameters and relative depth of points in space. The second approach treats the image flow field as a whole (Koenderink and van Doorn 1976; Longuet-Higgins and Prazdny 1980; Waxman and Ullman 1983; Wohn 1984; Waxman and Wohn 1985) in an attempt to recover the rigid body motion parameters and surface descriptions (slopes and curvatures) of entire surface patches. Recently, work has begun on the 3-D recovery of structure from non-rigid body motions (Ullman 1983; Koenderink, private communication). Our formulation of binocular image flows will follow the continuous field approach developed for monocular flows generated by textured objects in rigid body motion (Waxman and Ullman 1983, Waxman 1984, Waxman and Wohn 1984, Wohn 1984).

We consider a scene as comprised of objects in independent rigid body motion with respect to the observer. The individual objects are imagined as decomposed into surface patches visible to the observer, and these surface patches in space project into neighborhoods in the image. It is actually the surface texture and shading which is observed under perspective projection in the image. Due to the relative motion between object and observer, the projected texture undergoes deformations which reflect the image flow field. The theory of monocular image flows, developed by Waxman and collaborators (cf. References), provides techniques for the recovery of flow fields and deformation parameters from evolving contours, edge fragments and feature points in the imagery, and for recovery of 3-D surface structure and rigid body motion from these deformations. As these ideas provide the starting point for binocular flow analysis, they are reviewed in more detail here.

2.1 Image Velocity Relations

As a textured, rigid object moves through space, the evolving image sequence registered by a monocular observer (e.g. a moving pin-hole camera) contains information in the form of an image flow field. This image flow is determined by the relative rigid body motion between object and observer, as well as the structure of the object's surface visible to the observer. Derivation of this flow field follows that of Waxman and Ullman (1983).

We attribute the relative rigid body motion to an observer represented by the spatial coordinate system (X, Y, Z) in Figure 1. The origin of this system is located at the vertex of perspective projection, and the Z-axis is directed along the center of the instantaneous field of view. The instantaneous rigid body motion of this coordinate system is specified in terms of the translational velocity $V = (V_X, V_Y, V_Z)$ of its origin and its rotational velocity $\Omega = (\Omega_X, \Omega_Y, \Omega_Z)$. The 2-D image sequence is created by the perspective projection of the object onto a planar screen oriented normal to the Z-axis. The origin of the image coordinate system (x, y) on the screen is located in space at $(X, Y, Z) = (0, 0, 1)$; that is, the image is reinverted and scaled to a focal length of unity.

Due to the observer's motion, a point P in space (located by position vector R) moves with a relative velocity $U = -(V + \Omega \times R)$. At each instant, point P projects onto the screen as point p with coordinates $(x, y) = (X/Z, Y/Z)$. The corresponding image velocities of point p are $(v_x, v_y) = (\dot{x}, \dot{y})$, obtained by differentiating the image coordinates with respect to time and utilizing the components of U for the time derivatives of the spatial coordinates of P. The result is

$$v_x = \left\{ x\frac{V_Z}{Z} - \frac{V_X}{Z} \right\} + \left[xy\,\Omega_X - (1 + x^2)\Omega_Y + y\,\Omega_Z \right], \tag{1a}$$

$$v_y = \left\{ y\frac{V_Z}{Z} - \frac{V_Y}{Z} \right\} + \left[(1 + y^2)\Omega_X - xy\,\Omega_Y - x\,\Omega_Z \right] \tag{1b}$$

These equations define an instantaneous image flow field, assigning a unique 2-D image velocity v to each direction (x, y) in the observer's field of view. For the moment, we shall consider only a single surface patch of some object in the field of view. A small but finite surface patch may be locally approximated by a quadric surface in space as described by six parameters: two slopes, three curvatures and an overall distance scale. If the surface patch is described in this viewer-centered spatial coordinate system by $Z = \zeta(X, Y)$, then it is straightforward to find the corresponding local representation

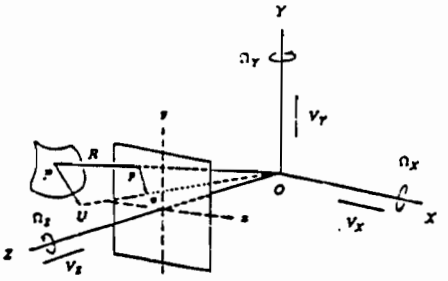


Fig. 1 - Spatial Coordinates Moving with a Monocular Observer and the Monocular Image Coordinates

$Z = Z(x, y)$ as a second-order polynomial in terms of image coordinates. Of these six surface parameters, only five can be recovered directly from the image flow field; *the overall scale factor is lost as it always appears in ratio with the translational velocity* V (Waxman and Ullman 1983). Moreover, the remaining five surface parameters appear in product with the translational space motion. The *kinematic analysis* developed by Waxman and Ullman (1983) leads to a set of twelve algebraic equations relating this 3-D structure and motion to derivatives (through second order) of the image flow. *Recovery of the 3-D information requires solution of nonlinear equations.*

2.2 Second-Order Image Flows

In the recovery of surface structure and 3-D motion from image flow, it is sufficient to describe an image flow as a locally second-order flow field. This has implications with regard to the surfaces which generate the flow itself. For example, a planar surface patch $Z = Z_0 + pX + qY$, may be described exactly as $Z = Z_0(1 - px - qy)^{-1}$ in image coordinates. Substitution into the velocity equations above yields expressions in the form of second-order polynomials. For planar surfaces, such second-order flows are *globally valid*. On the other hand, quadric surfaces generate flows which are not simple polynomials in the image coordinates. However, they may be *locally approximated* as second-order flows. The coefficients of this second-order flow then determine the slopes and (scaled) curvatures of the quadric surface patch as well as its (scaled) space motion. In this context, a complex surface is viewed as a composite of overlapping planar and quadric patches. The image flow associated with a smooth surface is, therefore, a slowly varying (in terms of image coordinates) second-order flow defined over a region of the image.

In order to recover the second-order flow approximation for any neighborhood in the image, it is necessary to have a sufficiently dense texture present in that neighborhood. This texture gives rise to extended contours, edge fragments and point features, all of which are convected

along and deformed by the local image flow. These features serve to sample components of the flow field; in particular, the contours and edges yield an estimate of the flow in the direction normal to the contours themselves. The *Velocity Functional Method* (Waxman and Wohn 1984) may then be used to recover the local flow from these sampled components.

We model the components of the local velocity field by second-order polynomials; hence, define the partial derivatives of image velocity evaluated at a local origin as

$$v^{(i,j)} \equiv \left. \frac{\partial^{i+j} v}{\partial x^i \partial y^j} \right|_0 . \qquad (2)$$

Then the components of instantaneous velocity in the neighborhood are described by the two functionals

$$v_x(x,y) = \sum_{\substack{i=0 \\ (i+j\le 2)}}^{2} \sum_{j=0}^{2} v_x^{(i,j)} \frac{x^i}{i!} \frac{y^j}{j!} , \qquad (3a)$$

$$v_y(x,y) = \sum_{\substack{i=0 \\ (i+j\le 2)}}^{2} \sum_{j=0}^{2} v_y^{(i,j)} \frac{x^i}{i!} \frac{y^j}{j!} . \qquad (3b)$$

The polynomial coefficients can be obtained by fitting the equations to the measured velocity at isolated feature points or to the flow normal to image contours, Waxman and Wohn (1984).

2.3 Boundaries of Analyticity

From equations (1) it is apparent that the flow field is "functionally analytic" (i.e. twice differentiable) wherever object surfaces $Z(x,y)$ are twice differentiable. The flow is non-analytic at points where Z or its first partials are discontinuous, and where the relative space motion parameters change. Such points occur along occluding boundaries and structural edges where surface orientation changes abruptly (e.g., the edges of a polyhedron). Thus, an image flow field is naturally partitioned into regions of analyticity separated by singular contours (i.e., *boundaries of analyticity*). These analytic regions are, in turn, decomposed into neighborhoods in which the image flow is locally approximated as a second-order flow. It is part of a complete image flow analysis to delineate these boundaries of analyticity so that 3-D interpretations can be assigned to the regions within them. Figure 2 illustrates this partitioning of the image flow field.

In order to detect the presence of a boundary of analyticity in the flow field, we try to "analytically continue" the flow from one neighborhood to the next. This is accomplished by requiring the separate second-order flow approximations determined in each neighborhood to be "compatible" in an overlapping area common to both neighborhoods (Wohn 1984; Wohn and Waxman 1985b). The degree of compatibility between neighboring flow approximations is measured relative to the agreement

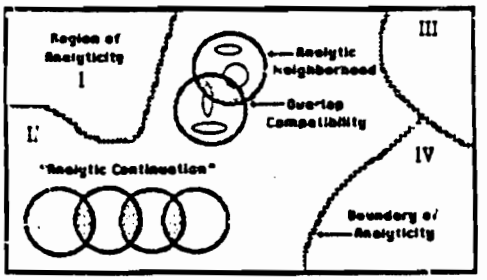


Fig. 2 - Partitioning the Velocity Image into Analytic Regions Seperated by Boundaries of Analyticity. Analytic Regions are Comprised of Overlapping Neighborhoods in which the Flow Field is Locally Second Order.

between the individual approximations and the data from which they are obtained. When neighboring flow approximations are deemed "incompatible," it is assumed that a boundary of analyticity has been crossed. This necessitates the splitting and merging of neighborhoods in order to localize this discontinuity. The beginnings of a control structure governing the automatic segmentation of flow fields is presented in Section 4 below.

2.4 Monocular Analysis of Binocular Flows

In the case of a binocular image sequence, the monocular flow analysis described above is to be applied to the left and right image sequences separately. But rather than going so far as the 3-D inference from monocular flow (Waxman and Ullman 1983) for each sequence, we consider only the recovery and segmentation of the separate image flows. This segmentation into analytic regions (i.e., regions of slowly varying second-order flow) allows gross correspondence to be established between these regions in the left and right images. It also delineates the depth and orientation discontinuities which often plague stereo matching and surface reconstruction algorithms.

This completes *Step 1* of our stereo-motion fusion module. The reconstructed flow fields for the left and right images are brought together in the stage of "binocular flow analysis" described next.

3. BINOCULAR IMAGE FLOWS

For simplicity, we restrict our analysis to the parallel stereo configuration illustrated in Figure 3. The left and right image planes lie in a common plane with the fixation point located at infinity (i.e., the "eyes" point straight ahead). The left and right coordinates, (x_l, y_l) and (x_r, y_r) respectively, have their origins at the centers of their respective fields of view separated by a baseline of magnitude b along the common direction of the x-axes. Each image plane is positioned at a focal length of unity with respect to a pin-hole located at the vertex of projection for each separate camera/eye. *This stereo*

configuration is assumed to move rigidly with respect to other moving objects in the scene. No allowance has been made for vergence of the eyes (known or otherwise) in the current formulation.

Consider the monocular flow analysis of *Step 1* already performed separately on the left and right image sequences. The analytic flow regions bounded by flow discontinuities are assumed to be brought into correspondence rather easily. This can be accomplished essentially by matching the flow discontinuities between left and right images. The correspondence is gross, but allows the binocular flow analysis to focus attention on individual regions. Each such region is assumed to correspond to a smooth surface of a rigid body. Thus, we may associate with each region a set of relative rigid body motion parameters. However, for the sake of analysis, if we ascribe the rigid body motion to the "monocular observer", as in Figure 1 and equations (1), then the rigid body motion parameters for a given region are different for the left and right cameras/eyes. This is due to the fact that the left and right cameras/eyes are in motion with respect to each other when relative motion between object and observer is ascribed to the observer. If according to the left coordinate system the rigid body motion parameters of a region are (V_l, Ω_l), then in the right coordinate system that same region has motion parameters (V_r, Ω_r), where

$$\Omega_r = \Omega_l \tag{5a}$$

$$V_r = V_l - \Omega_l \times b\hat{i}, \tag{5b}$$

and $\hat{i}$ is a unit vector in the common x-direction.

Thus, the image flow fields of the two eyes/cameras differ in magnitude as well as distribution (due to stereo disparity). And as both stereo disparity and monocular flow vary inversely with depth, we should not be surprised that binocular flow and disparity are related in a simple way. In fact, we shall see that binocular flow is synonymous with "rate-of-change of disparity."

3.1 Relative Flow - Disparity Relation

Given the parallel stereo configuration, we have the simple case of corresponding features lying along horizontal epipolar lines. Thus, a feature located at position (x_l, y_l) in the left image at some instant of time is located at (x_r, y_r) in the right image, where

$$y_r = y_l, \tag{6a}$$

$$\delta(x_l, y_l) \equiv x_r - x_l = b / Z_l(x_l, y_l), \tag{6b}$$

$\delta(x_l, y_l)$ being the angular disparity between right and left image positions of the feature at (x_l, y_l) in the left image. Note that over a particular analytic flow region, the (horizontal) disparity forms an analytic scalar field generated by the smooth depth function $Z_l(x_l, y_l)$. And since the left and right coordinate systems are parallel,

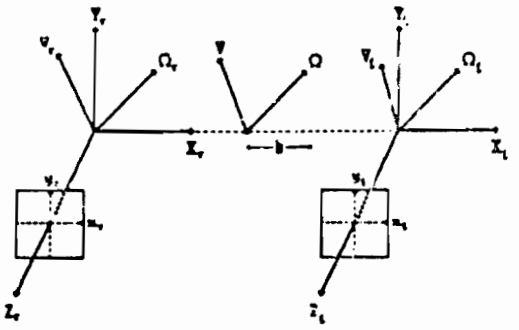


Fig 3 - Spatial and Image Coordinates for the Binocular Configuration. Space Motions are Show for Left, Right and Cyclopean Systems which Move a Rigid Object.

the depth function for the corresponding region in the right image may be expressed as

$$\begin{aligned} Z_r(x_r, y_r) &= Z_r(x_l + \delta[x_l, y_l], y_l) \\ &= Z_l(x_l, y_l). \end{aligned} \tag{7}$$

Let us rewrite the monocular image velocity relations (1) in terms of translation and rotation coefficient matrices,

$$v(x,y) = \frac{1}{Z(x,y)} \vec{T}(x,y) \cdot V + \vec{R}(x,y) \cdot \Omega; \tag{8}$$

these 2×3 matrices being functions of image coordinates alone with elements easily obtained from relations (1). Now an expression like (8) may be associated with each image in our stereo configuration; the coordinates, motion parameters and depth function are, however, different. In order to relate the left and right image flows for a given region, we shall express both flows in terms of the left coordinate system by using expressions (5,6,7). Thus, the left image flow is given by

$$v_l(x_l, y_l) = \frac{1}{b} \delta(x_l, y_l) \vec{T}(x_l, y_l) \cdot V_l + \vec{R}(x_l, y_l) \cdot \Omega_l, \tag{9a}$$

while the right image flow is given by

$$\begin{aligned} v_r(x_l + \delta, y_l) = \frac{1}{b} \delta(x_l, y_l) \vec{T}(x_l + \delta, y_l) \cdot \left\{ V_l - \Omega_l \times b\hat{i} \right\} \\ + \vec{R}(x_l + \delta, y_l) \cdot \Omega_l \end{aligned} \tag{9b}$$

Equations (9a,b) yield the image velocities of corresponding features in the two cameras/eyes.

Now we define the "relative flow" (or *binocular difference flow*) of features between the left and right images as the difference between the "shifted flow fields", the "shift" being associated with the disparity field;

$$\Delta v(x_l, y_l; \delta) \equiv v_r(x_l + \delta[x_l, y_l], y_l) - v_l(x_l, y_l). \tag{10}$$

Upon expanding the coefficient matrices of (9a,b) according to equations (1), forming the relative flow (10) and simplifying yields the following expressions for the components of relative flow;

$$\Delta v_x(x_l, y_l; \delta) = \frac{1}{b} V_Z \delta^2 + (y_l \Omega_X - x_l \Omega_Y)\delta , \qquad (11a)$$

$$\Delta v_y(x_l, y_l; \delta) = 0 . \qquad (11b)$$

Forming the ratio of relative flow to disparity yields

$$\frac{\Delta v_x(x_l, y_l; \delta)}{\delta(x_l, y_l)} = \frac{1}{b} V_Z \delta + (y_l \Omega_X - x_l \Omega_Y) , \qquad (12a)$$

$$\frac{\Delta v_y(x_l, y_l; \delta)}{\delta(x_l, y_l)} = 0 . \qquad (12b)$$

We shall interpret expressions (12a,b) momentarily. But first note that this ratio of relative flow to disparity is linear in the variables x_l, y_l and δ, with coefficients proportional to the unknown parameters of relative motion. The reader may verify for himself that, when reexpressed in the cyclopean coordinate system (midway between the two cameras/eyes), expressions (12) remain unchanged. Thus, we may suppress the subscript "l" in (12) and write instead,

$$\frac{\Delta v_x(x, y; \delta)}{\delta(x, y)} = \frac{1}{b} V_Z \delta(x, y) + (y \Omega_X - x \Omega_Y) , \qquad (13a)$$

$$\frac{\Delta v_y(x, y; \delta)}{\delta(x, y)} = 0 , \qquad (13b)$$

with image coordinates and motion parameters corresponding to the cyclopean coordinate system.

If we consider the relative flow in a small enough neighborhood such that the underlying surface patch may be treated as locally planar, then we have a simple expression for the local disparity field,

$$\delta(x, y) \equiv \frac{b}{Z(x, y)} = \frac{b}{Z_0}(1 - px - qy) , \qquad (14)$$

where Z_0 is the depth to the plane measured along the center of the cyclopean field of view, and p and q are the components of local slope. Substituting (14) for the disparity on the right-hand side of (12) yields the local relative flow to disparity relations,

$$\frac{\Delta v_x(x, y)}{\delta(x, y)} = \frac{V_Z}{Z_0} - \left(\frac{V_Z}{Z_0} p + \Omega_Y \right) x - \left(\frac{V_Z}{Z_0} q - \Omega_X \right) y \qquad (15a)$$

$$\frac{\Delta v_y(x, y)}{\delta(x, y)} = 0 \qquad (15b)$$

We see that locally, *the relative flow to disparity ratio is a linear function of image coordinates* with coefficients depending on the surface structure and relative motion between object and observer.

3.2 Interpreting the Correlation

The correlation between relative flow Δv and disparity δ, presented in cyclopean coordinates in (13a,b) is simple to interpret. Recall that we are considering only a parallel stereo imaging geometry, hence, the epipolar lines are horizontal (i.e., parallel to the x-axes). Now the relative flow Δv represents the rate of separation of a feature in one image, from its match in the other image. It is the rate of change of vector disparity. As a feature and its match must always lie along some epipolar line, its vertical disparity must remain zero in this case. Thus, relation (13b) expresses the fact that a feature and its match must flow perpendicular to epipolars at the same rate in order to lie on a common epipolar. In general, the rate of change of vertical disparity must be such as to keep a feature and its match on an epipolar line.

For our parallel stereo configuration, we may then identify Δv_x with the rate of change of (horizontal) disparity and denote it by $\dot{\delta}$. Returning to expression (14) we have

$$\dot{\delta} = -\frac{b}{Z^2} \dot{Z} = -\delta \frac{\dot{Z}}{Z} . \qquad (16)$$

From $U = -(V + \Omega \times R)$ we have $\dot{Z} = -V_Z - \Omega_X Y + \Omega_Y X$, hence,

$$\frac{\dot{Z}}{Z} = -\frac{V_Z}{Z} - (y \Omega_X - x \Omega_Y) = -\frac{V_Z}{b}\delta - (y \Omega_X - x \Omega_Y) . \qquad (17)$$

Combining (17) with equation (16) yields for $\dot{\delta}/\delta$,

$$\frac{\dot{\delta}(x, y)}{\delta(x, y)} = \frac{V_Z}{b} \delta(x, y) + (y \Omega_X - x \Omega_Y) , \qquad (18)$$

which is identical with relation (13a). *Thus, this correlation between relative image flows and stereo disparity is, in fact, a relationship between disparity and its rate of change.*

3.3 Using the Correlation to Establish Stereo Correspondence

If the relative motion between the cameras and the objects in the field of view is known, the binocular flow relations can be used to establish stereo correspondence directly. To do this, the correspondence of potential matches is tested by substituting the measured velocities

and positions of a pair of points into Equations (13). This technique is described in more detail in Waxman and Duncan (1985). If the relative motion parameters are unknown, it is possible to establish correspondence using a local support technique with the linearized version of the binocular flow relations (15). This later technique is similar to that suggested by Prazdny (1984) for use in stereo matching with static images.

4. EXPERIMENTS

A limited experimental program was undertaken to demonstrate the feasibility of implementing the first three steps of the stereo-motion module: *Step 1* (flow recovery and segmentation), *Step 2* (establishing correspondence using the binocular difference flow) and, to a limited extent, *Step 3* (recovering surface structure). A brief description of the experiments is given below. The interested reader is referred to Waxman and Duncan (1985) if more detail is desired. Binocular image flow fields were obtained using a camera mounted on a robot arm, viewing scenes consisting of white objects covered by black dots. In general, the experiments were successful insofar as they confirmed the potential of *overlap compatibility* for segmentation of laboratory flow data, and verified the binocular difference flow-disparity relations for a particular configuration. Still, much work remains before a fully automatic module is realized.

4.1 Apparatus and Procedures

The moving pair of stereo cameras was simulated using a single, black and white, Sony (model DC-37) CCD-camera mounted on an American Robot, MERLIN robot arm. The images were digitized into 480 × 420 pixel arrays using a Grinnell (GMR-27) display processor and memory. Throughout this section, all angular measurements are given in units of pixels; time is in units of seconds. Each image flow field was obtained from three frames taken with the camera at three positions, equally spaced in time, on its trajectory. The trajectories and viewing directions were chosen to simulate a pair of cameras in a parallel stereo configuration (cf. Fig. 3). The baseline between cameras was 3.0 inches.

The scenes consisted of white surfaces covered with a distribution of 0.125 inch diameter black dots. From the typical viewing distance of 40 inches the dots appeared in the image with a diameter of 3 pixels. The centroids of the dots were tracked for three frames and velocities at the centroids in the central frame in time were computed.

4.2 Image Flow Segmentation

We have analyzed the scene shown in Figure 4, which consists of a planar background with two connected planar surfaces in the foreground. The effective camera motions, also shown in the figure, were 0.25 inches/sec in the viewing direction (toward the scene) and

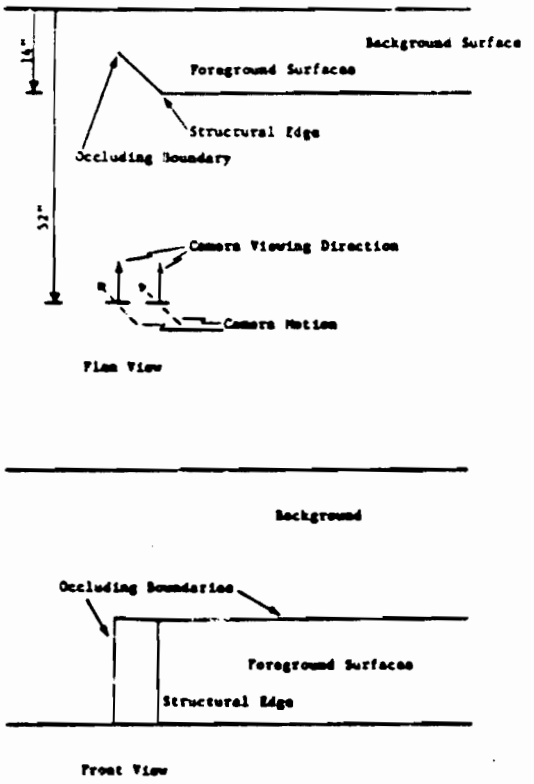


Fig. 4 - Two Views of the Scene Used for the Segmentation Experiments.

0.25 inches/second in the X-direction (parallel to the scene). At the central frame the cameras were about 40 inches from the foreground surfaces.

The current segmentation program reveals the potential locations of flow discontinuities, but does not refine them nor link them into global boundaries of analyticity. The program first divides the image into N^2 equal-sized rectangles; in this case, a 5 × 5 rectangular grid on each 480 × 420 pixel image. Each rectangle contained an average of about 10 feature points. The velocity data in each rectangle was then fit to a pair of second-order polynomials (cf. equations 3) using a linear least squares approach. The error per point between the data and the second order fit, defined as

$$err = (N \mid v_{avg} \mid)^{-1} \sum_{i=1}^{N} \left[\left| v_x'|_{poly} - v_x'|_{meas} \right| + \left| v_y'|_{poly} - v_y'|_{meas} \right| \right] \quad (24)$$

was typically 0.02 .

In an attempt to see if the polynomial flow fields from adjacent rectangles were compatible, i.e., belonged to the same analytic flow region, the velocities were compared in overlapping neighborhoods. Specifically, at vertical boundaries between left and right rectangles and at

horizontal boundaries between upper and lower rectangles, an overlap compatibility measure (C_v and C_h, respectively) was computed,

$$C_v = \frac{2.0}{(err_r + err_l)} \left[\frac{1}{A_v} \cdot \iint_{A_v} (v_r - v_l)^2 dxdy\right]^{1/2}, \quad (25a)$$

$$C_h = \frac{2.0}{(err_t + err_b)} \left[\frac{1}{A_h} \iint_{A_h} (v_t - v_b)^2 dxdy\right]^{1/2}, \quad (25b)$$

where A_v and A_h are areas around the vertical and horizontal boundaries respectively. The velocities v_r and v_l refer to the velocity functionals from the right and left sides of vertical boundaries, while v_t and v_b refer to the velocity functionals from the top and bottom sides of horizontal boundaries. After computing the compatibility for the original 5 × 5 rectangular grid, the calculations were repeated twice with the grid shifted to the right in each case by one-third the rectangle width (approximately the distance between feature points). The three horizontal grid positions were then repeated with the grid shifted down by one-half the rectangle height. Thus, the overlap error was computed for boundaries of 6 rectangular grids with 25 rectangles in each grid. A plot of the overlap compatibility function is shown in Figure 5 for the vertical boundaries of the left image. A similar plot for the horizontal boundaries appears in Figure 6. Consider the compatibility across vertical boundaries first, Figure 5. Note that the contours with $C_v = 4$ (i.e., four times the error in fitting the polynomials) do not correspond to any structural feature of the scene. Thus, the noise level appears to be about 4. In Figure 5, both the vertical occluding boundary and the vertical structural edge appear in the contours with compatability errors as high as 10, i.e., 2.5 times the noise level. For the structural edge (i.e., the slope discontinuity) the largest values appear slightly to the right of the feature. Note that these contours also indicate, to some extent, the position of the horizontal occluding boundary. This horizontal boundary is seen more clearly in the compatibility of upper-lower pairs of rectangles, Figure 6. The compatibility function is again typically 8 to 10 at the boundary.

The flow field segmentation results indicate that the overlap compatibility method can sucessfully locate occluding boundaries (i.e., depth discontinuities) and to some extent structural edges (i.e., slope discontinuities) in real data. However, the noise level and resolution of the results need to be improved. It is believed that both of these problems can be remedied by increasing the density of data points in the images, Waxman and Duncan (1985).

4.3 Binocular Flow Field Experiments

In this section we describe a preliminary experimental exploration of the binocular flow equations (11). In particular, a V_Z motion was chosen for the camera pair and the equations were verified. It was pointed out in

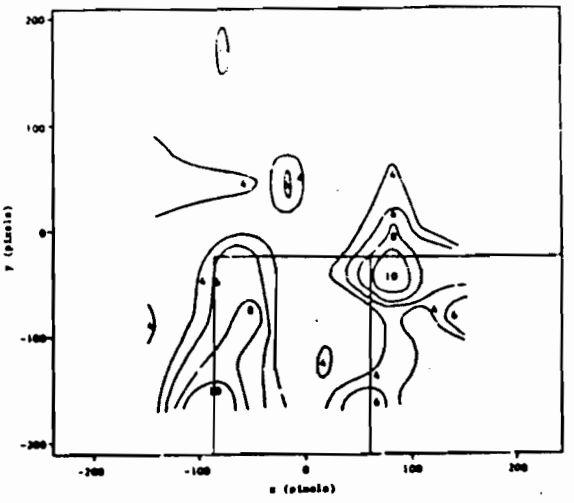


Fig. 5 - Overlap Compatibility Contours Across Vertical Boundaries. C_v - Left Image.

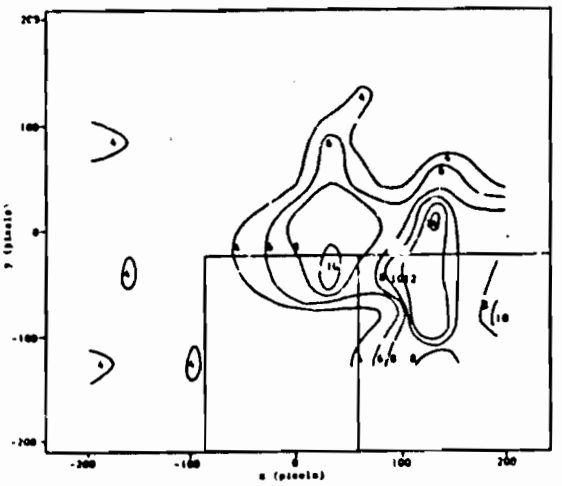


Fig. 6 - Overlap Compatibility Contours Across Horizontal Boundaries. C_h - Left Image.

Waxman and Duncan (1985) that the V_Z motion is one of the two single component motions that will allow accurate discrimination between correctly and incorrectly matched features. The experiment used the camera set-up described earlier to simulate a pair of cameras separated by a 3 inch baseline. The cameras viewed a planar surface perpendicular to the viewing direction (i.e., a frontal plane). The velocity fields were obtained from three images with the cameras at 43.5, 45.0 and 46.5 inches from the surface.

The binocular flow equations (11) were verified by two techniques: one using the individual data points and the other using the polynomial fits to the velocity fields;

the space motion being known in both cases here (which is not generally true). In general both methods proved successful in this preliminary experimental program. The details are given in Waxman and Duncan (1985).

6. CONCLUSIONS

In this paper we have outlined a set of five steps toward the development of a stereo-motion fusion module. The successful development of a complete module of this type has enormous potential for robotics in a dynamic environment. It may also shed some light on the nature of the processing going on in the human visual pathway. In this respect, the work of Regan and Beverley (1979) is most relevant, for their own psychophysical and neurophysiological studies have led them to suggest the existence of neural organizations which may "compute" the binocular difference flow (or relative flow between the eyes) which is so basic to our own theory.

The basic advantages this module offers over static stereo are: monocular detection of the depth and orientation discontinuities (before matching is attempted), use of a correlation between binocular difference flow and disparity to drive the matching process (either independent of, or in conjunction with matching based on disparity alone), the ability to refine disparity estimates to sub-pixel accuracy by considering the smooth orbits of features through the left and right image space-times, and the potential to focus attention of the matching process to the areas where new features enter the field of view. The advantages of this module over structure from monocular motion are: the ability to recover absolute structure and rigid body motions (without scale factor ambiguities), and that only linear equations need be solved to recover rigid body motion parameters.

Still, much work remains to be done before a complete module of this type can be constructed. The control structure for the flow segmentation procedure requires further development. This segmentation procedure should be iterative, with subsequent refinements occurring near detected flow discontinuities. The discontinuities in left and right images must also be matched in order to establish gross correspondence among analytic regions. The binocular difference flow-disparity relation, derived in Section 3, requires further testing in order to insure its validity under more general classes of motion than tried here. It should also be generalized to incorporate vergence effects. The matching techniques described in Section 5 need to be implemented and tested in a variety of cases. The ability to combine evidence in establishing correspondence is an appealing aspect of the approach and needs to be implemented as well.

The possible role of a combined stereo-motion module, such as this one, in the human visual processing task raises some interesting questions. How does the brain utilize disparity estimates and binocular flow-disparity cues in establishing correspondence? Does one take priority over the other, or are they combined? What happens when structure from binocular flow conflicts with structure from static stereo (Mayhew and Frisby, private communication)? Does one percept dominate or do we see illusions? Are there certain kinds of "head motions" preferred for disambiguating false matches? Is there a "gradient limit" effect associated with the coefficients of the linear terms in equation (15a)? Is it possible to fuse a dynamic stereogram which is beyond the static disparity gradient limit of unity? Perhaps psychophysical experiments can resolve some of these questions.

REFERENCES

Adiv, G. 1984 (October). Determining 3-D motion and structure from optical flow generated by several moving objects. *Proc. DARPA Image Understanding Workshop*, New Orleans: SAIC, pp. 113-129.

Burt, P. and Julesz, B. 1980. A disparity gradient limit for binocular fusion. *Science*, 208: 615-617.

Buxton, B.F., Buxton, H., Murray, D.W. and Williams, N.S. 1984. 3-D solutions to the aperture problem. *European Conf. Artificial Intelligence '84*.

Eastman, R. and Waxman, A.M. 1985. Using disparity functionals for stereo correspondence and surface reconstruction. Tech. Report in preparation. College Park, MD: University of Maryland, Center for Automation Research.

Grimson, W.E.L. 1981. *From Images to Surfaces*. Cambridge: M.I.T. Press.

Jenkin, M.R.M. 1984 (September). The stereopsis of time-varying images.
Tech. Report RBCV-TR-84-3. Toronto, Canada: University of Toronto, Dept. of Computer Science

Koenderink, J.J. and van Doorn, A.J. 1975. Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. *Optica Acta* 22. 773-791.

Koenderink, J.J. and van Doorn, A.J. 1976. Geometry of binocular vision and a model for stereopsis. *Biol. Cybernetics* 21: 29-35.

Longuet-Higgins, H.C. 1981. A computer algorithm for reconstructing a scene from two projections. *Nature* 293: 133-135.

Longuet-Higgins, H.C., and K. Prazdny, K. 1980. The interpretation of a moving retinal image. *Proc. Roy. Soc. Lond.* B208: 385-397.

Marr, D. 1982. *Vision*. San Francisco: Freeman.

Marr, D. and Poggio, T. 1979 A computational theory of human stereo vision. *Proc. Roy. Soc. Lond.*, B204: 301 328.

Mayhew, J.E.W. and Frisby, J.P. 1981. Psychophysical and computational studies towards a theory of human stereopsis. *Artificial Intelligence* 17: 349-385.

Poggio, G.F. and Poggio, T. 1984. The analysis of stereopsis. *Ann. Rev. Neurosci.*, 7: 379-412.

Pollard, S.B., Mayhew, J.E.W. and Frisby, J.P. 1985. Disparity gradients and stereo correspondences. Preprint, Dept. Psychology, Sheffield University.

Prazdny, K. 1980. Egomotion and relative depth map from optical flow.
Biol. Cyber. 36: 87-102.

Prazdny, K. 1984. Detection of binocular disparities. Preprint, Fairchild Laboratory for Artificial Intelligence Research. *Biol. Cyber.* (in press) 1985.

Regan, D. and Beverley, K.I. 1979. Binocular and monocular stimuli for motion in depth: Changing-disparity and changing-size feed the same motion-in-depth stage. *Vision Research* 19: 1331-1342.

Richards, W. 1983. Structure from stereo and motion. A.I. Memo 731. Cambridge, MA: Massachusetts Institute of Technology, Artificial Intelligence Laboratory. See also, *J. Opt. Soc. Amer.* A2: 343-349 (1985).

Subbarao, M. and Waxman, A.M. 1985. On the uniqueness of image flow solutions for planar surfaces in motion. Tech. Report 113. College Park, MD: University of Maryland, Center for Automation Research.

Tsai, R.Y. and Huang, T.S. 1981a. Uniqueness and estimation of 3-D motion parameters of rigid objects with curved surfaces. Report R-921. University of Illinois/Urbana-Champaign Coordinated Science Lab.

Tsai, R.Y. and Huang, T.S. 1981b. Estimating 3-D motion parameters of a rigid planar patch. Report R-922. University of Illinois/Urbana-Champaign Coordinated Science Lab.

Ullman, S. 1979. *The Interpretation of Visual Motion.* Cambridge: MIT Press.

Ullman, S. 1983. Maximizing rigidity: the incremental recovery of structure from rigid and rubbery motion. Memo 721. Cambridge, MA: Massachusetts Institute of Technology Artificial Intelligence Laboratory

Waxman, A.M. 1984 (April). An image flow paradigm. *Proc. 2nd IEEE Workshop on Computer Vision: Representation and Control,* Annapolis: IEEE, pp. 49-57.

Waxman, A.M. and Duncan, J.H. 1985 (May). Binocular Image Flows: Steps Toward Stereo - Motion Fusion. Tech. Report 119. College Park, MD: University of Maryland, Center for Automation Research.

Waxman, A.M. and Sinha, S. 1984 (October). Dynamic Stereo: Passive ranging to moving objects from relative image flows. *Proc. DARPA Image Understanding Workshop.* New Orleans: SAIC, pp. 130-136.

Waxman, A.M. and Ullman, S. 1983 (October). Surface structure and 3-D motion from image flow: A kinematic analysis. Tech. Report 24. College Park, MD: University of Maryland, Center for Automation Research. Also see *Int. J. Robotics Research* 4 (3), 1985.

Waxman, A.M. and Wohn, K. 1984 (April). Contour evolution, neighborhood deformation and global image flow: Planar surfaces in motion. Tech. Report 58. College Park, MD: University of Maryland, Center for Automation Research. Also see *Int. J. Robotics Research* 4 (3), 1985

Waxman, A.M. and Wohn, K. 1985. Contour evolution, neighborhood deformation and image flow: Textured surfaces in motion. *Image Understanding 1985,* (eds.) W. Richards and S. Ullman. Norwood: Ablex Publishing.

Wohn, K. 1984. A contour-based approach to image flow. Ph.D. Thesis, University of Maryland, Department of Computer Science.

Wohn, K. and Waxman, A.M. 1985a. Contour evolution, neighborhood deformation and local image flow: Curved surfaces in motion. Tech. Report in preparation. College Park, MD: University of Maryland, Center for Automation Research.

Wohn, K. and Waxman, A.M. 1985b. The analytic structure of image flows: Deformation and segmentation. Tech. Report in preparation. College Park, MD: University of Maryland, Center for Automation Research.

GC970145-391

Detecting Structure in Random-Dot Patterns

Richard Vistnes

Computer Science Department
Stanford University, Stanford, California 94305

Abstract

This paper presents the results of some psychophysical experiments whose purpose was to determine the parameters that affect the detectability of straight dotted lines and curved dotted lines, embedded in a surround of random dots. Results show that performance does not depend on line length or regularity of dot spacing, but rather on the ratio of dot spacing relative to surround, on target curvature, and on "jaggedness" of the target. Speculations are made about a mechanism that may be able to duplicate this performance, as well as perform two other important tasks in early vision.

Introduction

An important task in early vision is the detection of patterns in images. As Witkin and Tenenbaum (1983) pointed out in a recent paper, human beings can usually find the structures in images at a very early stage, before any semantic meaning has been attached to those structures. These structures are usually detected by what is known in the psychological literature as "pre-attentive vision" (see Neisser (1967), and Triesman (1985) for some recent work). Pre-attentive human vision is able to perform perceptual tasks, probably by using parallel machinery, in a very short time (under 200 milliseconds). Pre-attentive discriminations are made automatically, without focused attention. Pre-attentive vision can detect lines and curves in images; find instances of parallelism and symmetry; and segment images on the basis of a variety of "texture" differences such as intensity, size of elements, color of elements, and orientation of elements. For an example of what can be accomplished by pre-attentive vision and what cannot, consider Figure 3. The linear structure in (a) is immediately apparent, while the structure in (b) is not perceiveable without scrutiny — that is, without attention.

Pre-attentive vision is closely related to the phenomenon of perceptual organization in vision. Perceptual organization can be defined loosely as the process of forming descriptions of the significant structures in the image. Some important kinds of structure include symmetry, parallelism, linear structures, curved structures, etc. Human perceptual organization is usually preattentive, i.e., it occurs rapidly and in parallel across the visual field.

While studying the phenomenon of perceptual organization, I became interested in the question of how well human beings can perceive curvilinear structures when those structures are embedded in a noisy background. This question has practical import in computer vision as well as in human perception. In current theories of image understanding, one of the first steps in image processing is edge finding. Local edge operators are applied to the image; the result of this process is an array of "edgels" that indicate the presence of an edge at that point. These edgels then must be linked into lines and curves. This problem is difficult because there is typically noise surrounding the "correct" edgels. Lowe (1984, 1985) studied the problem of linking points into curvilinear structures. He used a statistical approach similar to that proposed by Witkin and Tenenbaum (1982) for finding unlikely patterns in images.

Kass and Witkin (1985) have done some work related to this problem. Their system analyzed oriented patterns such as the well-known Glass patterns, and found vector fields corresponding to the dominant flow in the pattern. Zucker (1982, 1983) used lateral inhibition among orientation-specific operators to estimate local orientation in dot patterns.

In order to better understand how people perform in this task, I performed a number of fairly careful (if tedious) psychophysical experiments in order to find the parameters that affect the detectability of certain patterns in images. This paper reports some preliminary results of those experiments. The patterns in which I was interested were dotted straight lines and dotted circular arcs embedded in a surround of random dots. It is useful to study dot patterns, since it removes any shape or intensity differences between the elements of the pattern and forces the visual system to use only textural segregation mechanisms. The experiments varied several parameters of the target pattern, such as the length, or spacing of dots, and measured subjects' accuracy in detecting the presence of the target. There was typically a threshold for detection below which the target could not be easily seen, and above which it could be detected accurately. In this study, I was not interested in finding the precise threshold values for the parameters that determine our ability to detect these patterns. Rather, I was interested in determining which parameters affect our performance, and the general range of values that determine the threshold.

In this paper, I first report the experimental setup and follow with the experiments and their results. I then make some speculations about how the problem of finding such patterns in noisy backgrounds could be solved, both by biological and machine vision systems. I present some preliminary results of an algorithm to find curvilinear patterns such as those considered here, and suggest that the same mechanism may be useful for solving two related problems in perceptual organisation.

Experimental Methods

In all these experiments, the images were displayed to subjects for a short time (about 200 msec). As I noted above, I am interested in pre-attentive detection of structure, not the structure that can be seen only upon prolonged inspection. The 200 msec time limit prevents prolonged inspection of the pattern, and also prevents the subject from moving the eyes to use foveal high resolution to inspect a part of the image.

In this paper, I present the results of experiments that test detection of two kinds of target: straight dotted lines and curved dotted lines, embedded in a background of random dots. The target appeared in the image in one of four positions, chosen at random: vertical on the right or left, or horizontal on the top or bottom. These positions are shown in Figure 1. The image was presented for a short time, after which the subject indicated (by a keystroke) where he or she thought the target had been. The computer accumulated the accuracy of detection over a number of trials.

We now look at the experimental parameters that control the appearance of the target and the surround in these displays. The appearance of the random surround can briefly be described (but see the details below) by one parameter, the average spacing between dots d_s (Distance in Surround).

Consider the parameters that affect the shape of the dotted target. First, a straight line of evenly-spaced dots has two parameters: the length L of the line and the spacing d_t of the dots. The number of dots in such a line is $\lfloor L/d_t \rfloor$. We can allow the dots to be unevenly spaced along the line, but still have the same average spacing d_t by specifying a longitudinal variation v_L. This parameter specifies the maximum variation in a direction along the line. Each dot in the line will be perturbed by an amount $0.5Rd_tv_L$ in either direction along the line away from its nominal (evenly-spaced) position, where R is a random number between -1 and 1. When v_L is zero, the dots are evenly-spaced.

We can allow the dots to be displaced away from a straight line (to form a jagged line) by specifying a transverse variation v_T. This specifies a maximum variation in a direction normal to the line. Each dot is perturbed by an amount $0.5Rd_tv_T$ in a direction normal to the line, again away from its nominal position; R is again a random number between -1 and 1. When v_T is zero, the dots fall in a straight line.

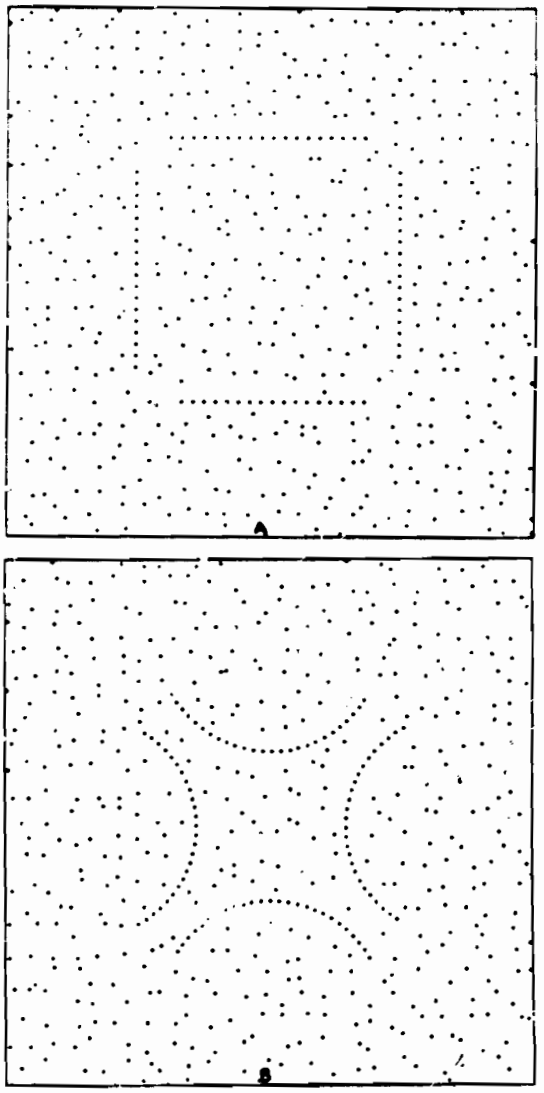


Figure 1. Positions of lines (a) and arcs (b) in images.

By specifying various values for these four parameters, many sorts of dotted lines can be generated. They all have the property that the average number of dots in the line per unit length (or the average density of dots) is uniform.

Now consider a dotted circular arc, such as the one shown in Figure 2. The shape of an arc of evenly-spaced dots can be specified by the length L of the chord joining the ends of the arc, the height H of the arc above this chord, the average spacing d_t of dots in the arc, and the transverse and longitudinal maximum variations v_T and v_L. To be precise, the dot spacing d_t is the linear distance between dots, rather than the distance along the arc joining them; that is, it is the distance along the chord joining the dots. The transverse variation is in the direc-

tion along the line through the dot and the center of the circle of which the arc is a part, i.e., along the diameter of the circle. The longitudinal variation is along a line normal to this diameter, that is, along the tangent to the circle. Note that when H is zero, an arc specified in this way is identical to a straight line as specified above.

Details

In this section I will present some of the details of the experimental setup and of the image-generating process. The casual reader need not read this, and may wish to skip to the next section.

Experimental setup. The images were generated by, and displayed on the screen of, a Symbolics 3600 Lisp Machine. In these experiments, the width of the image was 5 inches (460 pixels); this is equivalent to about 92 pixels per inch. The subject's eyes were about 17 inches away from the image, so the image subtended 15.7 degrees; the length of the line in most cases (200 pixels) was about 2 inches or 6.7°. The distance from the center of the display (and the fixation point) to the line was 1.2 inches or 4.0°.

The author was the subject for most of the experiments reported in this paper, since the experiments were so tedious that it was difficult to persuade others to take part. However, the services of two other subjects were obtained for a short time in order to verify that the data reported here is similar for different people.

Image generating. When I first generated images of dotted lines embedded in a surround of random dots, I noted that the dots in the surround often formed spurious groups (clusters and lines), simply due to random proximity of dots. This often proved distracting when trying to detect the dotted lines themselves. To avoid this problem, I used a method that generates more uniform (pseudo-random) patterns of dots.

The method works as follows. The algorithm is given an average separation d_s for the dots in the surround. The nominal position for each dot is at the intersection of grid lines d_s units apart. Such a pattern is completely regular. The position of each dot is then perturbed by independent amounts in the vertical and horisontal directions. The maximum amount of perturbation is specified by v_s; the maximum perturbation is $0.5Rd_sv_s$, where R is some random number in $[-1, 1]$. Another way of looking at this is that the dots are placed at random in a square with side d_sv_s. When v_s is zero, the dots are completely uniform; as v_s increases, the pattern looks more "random." When $v_s > 1$ the dots may touch, so for these experiments v_s was less than 1. When v_s is nearly 1, the patterns generated appear quite random, as evidenced by the surround of Figure 3.

Note that v_s was essentially an artifact of the process that generated the random surround; I did not consider it to be a parameter of the experiment. Therefore it was left constant at 0.8 in these experiments.

There is another subtlety in generating these images. When a dotted line is embedded in a surround of dots, it is possible for the dots in the line to fall close to dots in the surround. This creates local clusters of dots, and may trigger processes that we don't wish to study in this experiment. To avoid this problem, we need to make sure that there are no dots from the surround in the immediate vicinity of the dots in the line. To be precise, no dot should fall within a radius of $0.5d_s$ of any dot in the line. When generating images, we can either generate the surround dots first, then the dots in the line, and erase nearby dots; or else generate dots in the line first, then dots in the surround, and make sure not to put in the image any surround dot that is too close to a line dot. The latter approach was taken in these experiments.

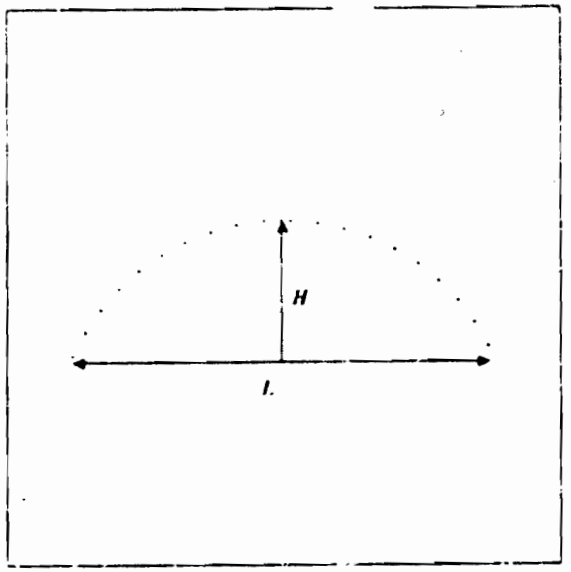


Figure 2. A typical circular arc. Chord length is L and height is H.

Experiments and Results

I was interested in finding the effect of all the parameters of the lines upon detectability. The experiments reported here attempted to isolate a single parameter and tested to see how subjects' performance varied, as measured by their accuracy in guessing the position of the target. Each of the experiments tested the effect of one parameter on detection ability.

I performed two primary experiments and several subsidiary experiments. The main experiments sought to determine the effect of deviation from a straight line, and the effect of curvature, on detection performance. In the former, a series of trials was presented in which the degree of alignment of the dots in the line varied; that is, v_T varied (recall that as v_T increases, so does the "jaggedness" of the line). As expected, performance dropped off as the deviation increased (details will be presented below). The trials were randomized so that the amount of deviation on a particular trial could not be predicted. In the latter experiment, a series of trials was presented in which dotted arcs of different curvatures appeared. The trials were again randomized so that the shape of the target could not be predicted. Again as expected, performance dropped off as curvature increased.

In the course of performing the experiments on arc detection, I noted that it was much easier to detect an arc when the shape was known in advance. That is, if the arcs presented in a series of trials all had the same L and

H, they all had the same shape. In this situation, it was possible to infer the presence of the arc from the presence of only a few dots in the right positions. On the other hand, when arcs of several different shapes were intermixed in the series of trials, they became more difficult to detect; i.e., the dots in the arc needed to be closer together than in the former case. Since I was interested in detection of structure without knowing the shape of that structure *a priori*, the shapes of the arcs were intermixed in these experiments.

The subsidiary experiments sought to determine how variables like length L and longitudinal variability v_L affected performance. These experiments received less emphasis, in terms of both experimental time and prominence in this paper, because they did not seem to be very important in determining our threshold.

Initial experiments showed that it is possible to detect, with nearly 100% accuracy, a straight-line target in a random surround approximately when $d_t \leq d_s$, if v_T was close to zero (i.e., $v_t < 0.2$). That is, straight dotted lines can be detected when the dots of which they are composed are closer together than those in the surround. (The reader may wish to look at Figures 3(a) and 3(b) and compare the ease with which he or she can detect the dotted lines.) In performing the experiments, I tried to see how altering some of the parameters affected performance. However, it was necessary to first set the other parameters so that the resulting target was near a threshold of detectability. That is, if $d_s = 20$ and $d_t = 10$, the dots in the target are very close compared to those in the surround, and varying another parameter such as H or v_L has very little effect on performance; it will be close to 100%. However, if we let $d_s = 20$ and $d_t = 20$ also, we are near the threshold for detection, so the effects of different H or v_L can be studied.

I will discuss the short, subsidiary experiments first. Then I will discuss the effects of transverse variation, followed by the effects of curvature, on performance.

Line length

I tested the detectability of lines of different L, for cases above and near the threshold of detectability. Some typical results are shown in Figure 4(a). Figure 4(b) plots performance as a function of the number of dots in the line, $\lfloor L/d_t \rfloor$. Note that the target can generally be detected when there are more than 5 or 6 dots in the line, if it can be detected at all (i.e., if a long target is above threshold). Moreover, accuracy does not seem to increase (once past the threshold) as the line length increases. This parameter, then, does not seem to be an important one for detection.

Longitudinal variation

Recall that this parameter v_L controls the variation of dot position along the line; as v_L increases the dots appear more irregularly-spaced. I tested the effect of increasing variation on accuracy of detection. Some results of this experiment are shown in Figure 5. This parameter does not seem to be crucial in determining detectability either.

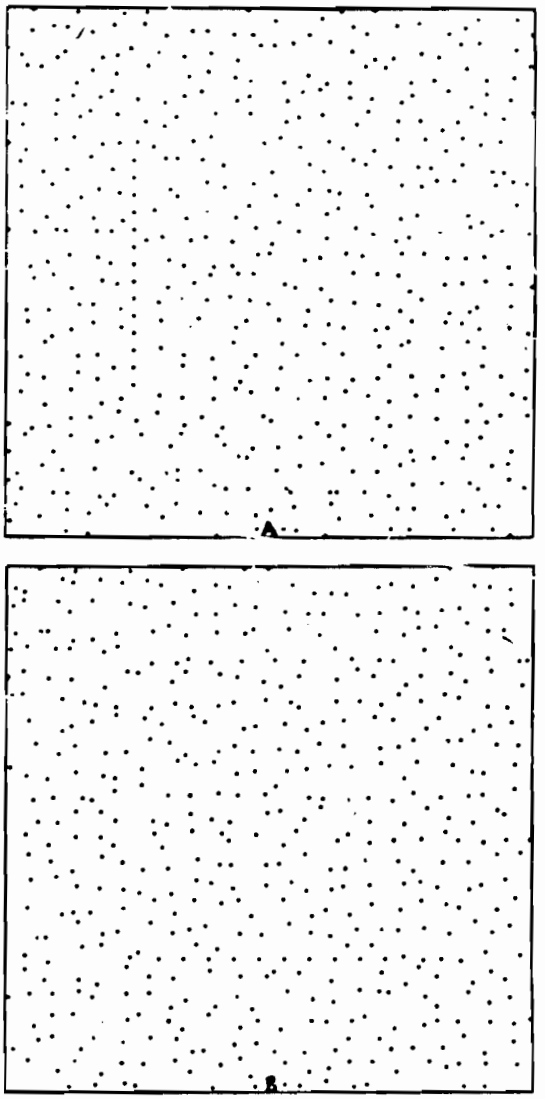
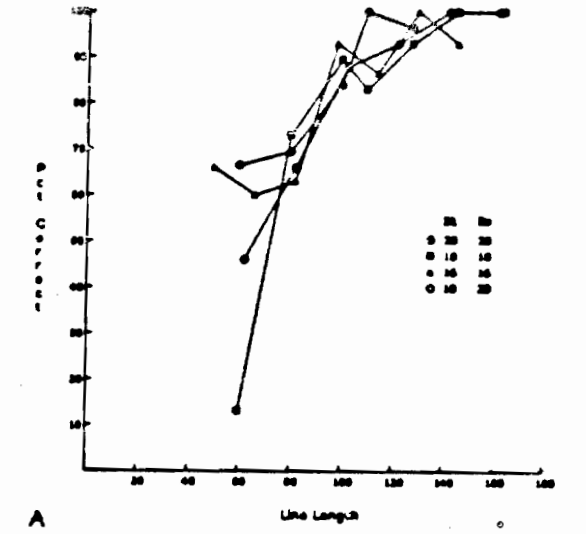


Figure 3. Comparison of difficulty of detecting straight-line patterns. In (a), the parameters are: $L = 200$, $d_s = 20$, $d_t = 15$, $H = 0$, $v_s = 0.5$, $v_T = 0$ and $v_L = 0$. In (b) the only difference is that $d_t = 22$; the target is difficult to detect.

Figure 5 shows that v_L has a slight effect on performance when the line is near the threshold of detectability, but in general it is safe to say that longitudinal variation does not matter much.

Transverse variation

In this experiment, the first of the primary experiments, I tested the effect of transverse variation v_T on accuracy of detection. These experiments were conducted with straight lines only; that is, $H = 0$. Starting with choices for the parameters that resulted in above-threshold targets, the transverse variation was increased until performance dropped to near the chance level (25%). This was



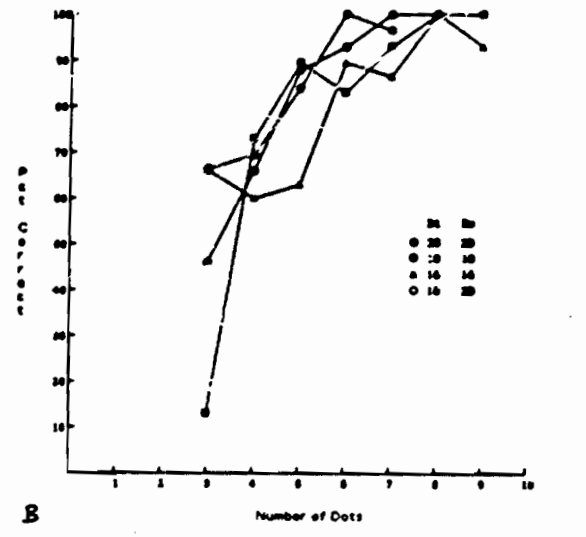


Figure 4. (a) Performance as a function of target length L. (b) Performance as a function of number of dots in the line.

repeated for various values of d_s and d_t; L and v_L were kept constant at 200 and 0.8, respectively, since those two parameters didn't seem to make much difference in the accuracy. Some typical results of this experiment are shown in Figure 6. Note that accuracy typically starts off at a plateau (100% if $d_t < d_s$), then drops as the threshold is crossed, and finally levels off at the chance level.

I asked what is the maximum v_T that allowed detection, for a particular setting of L, d_s, d_t etc. I chose a cutoff point of 80% accuracy to mark where performance dropped off, i.e., the threshold. For each setting of the parameters, then, I found the maximum v_T and plotted

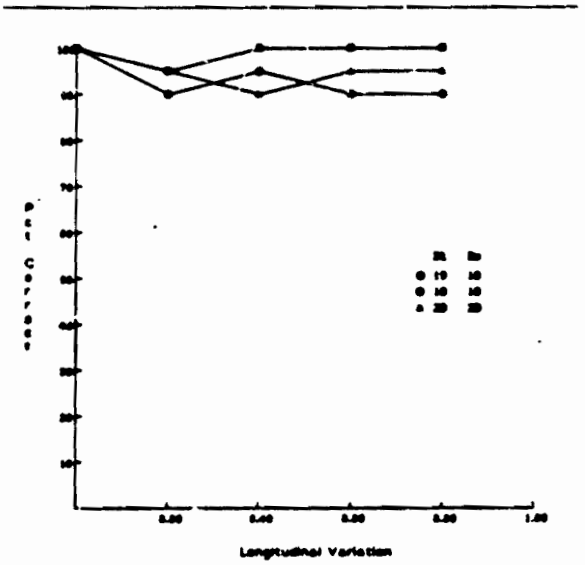


Figure 5. Performance as a function of longitudinal variation v_L.

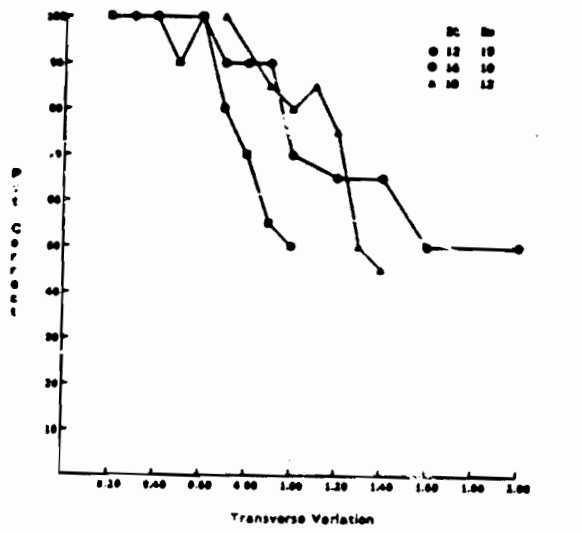


Figure 6. Typical results: Performance as a function of transverse variation v_T.

the results in Figure 7. Note that for constant d_s, as d_t increases, the maximum tolerable variation decreases; that is, as the distance between the dots in the target increases relative to the distance between dots in the surround, less jaggedness can be tolerated. This is in accord with our intuition.

Consider the relative separation of the dots in the target compared with those in the surround, d_t/d_s. If we plot maximum variation v_T versus relative separation, we get a good fit to a straight line, as shown in Figure 8. These

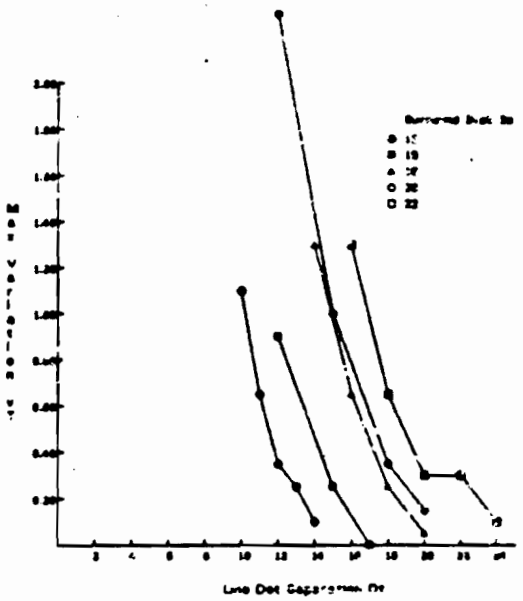


Figure 7. Maximum transverse variation v_T as a function of dot separation d_t, for various constant d_s.

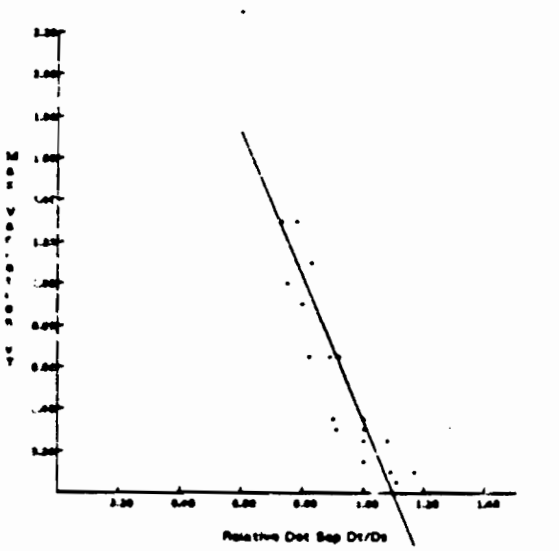


Figure 8. Maximum transverse variation v_T as a function of relative separation d_t/d_s. The result is linear to a good approximation; the slope of the regression line is -3.46 and the correlation coefficient is -0.92.

results indicate that it is only the relative separation, not the absolute separation, between the dots in the target that affects performance.

Height of arc (arc curvature)

In this experiment, the second of the primary experiments, I tested the effect of arc height, or curvature, on performance. I presented a variety of arcs of different curvature to the subject, each with the same L, d_t, d_s and variations v. I was interested in how performance varied with curvature. Recall that in this experiment, the curvatures were randomized, hence the subject could not predict the shape of the target; so the thresholds for detection are somewhat higher than in the previous experiments. They are also more general in the sense that we generally have no prior knowledge of the shapes we will need to organise. Some typical raw results of this experiment are shown in Figure 9.

I again used a criterion of 80% for the threshold of detection, and obtained the maximum H for each parameter setting; the results are plotted in Figure 10. I again determined the relative separation of dots in the arc and plotted maximum H versus d_t/d_s in Figure 11(a). The curvature κ of an arc with endpoint separation L and height H is

$$\kappa = \frac{2H}{(L/2)^2 + H^2}$$

The maximum curvature is plotted as a function of relative target separation in Figure 11(b). Note that the result is again linear to a fairly good approximation (correlation coefficient is -0.72). This again indicates that it is the relative separation, rather than absolute distance, that is important in detecting these kinds of dotted targets.

We can look at the results of this experiment another way. For a given arc height (or, equivalently, arc curvature), what is the farthest apart the dots in the arc can be for the arc to be detected? That is, what is the maximum d_t for a given H? I again used 80% as the threshold for detection and obtained the results in Figure 12(a), which shows the maximum d_t as a function of arc height H, for various surround separations d_s. Note that there is a point past which increasing the height has little effect on the maximum d_t. In Figure 12(b) I have plotted the maximum relative separation d_t/d_s as a function of arc curvature. We see the same trend here: once curvature increases past a certain point, the maximum separation does not change much. It appears that curvature has an increasingly detrimental effect on detectability when the curvature is small, but the effect stabilises when the curvature is large.

Conclusions

These experiments indicate that, to a good approximation, the length of the target, or number of dots, is not an important parameter in detection performance, as long as there are at least half a dozen dots in the target. Nor is the regularity of dot spacing along the line an important factor.

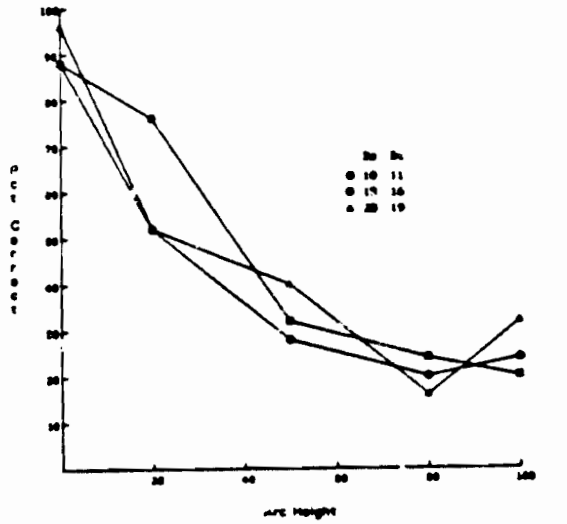


Figure 9. Typical results: accuracy of detection of arcs as a function of arc height H. Curves have constant d_t and d_s. All arcs have $L = 200$, $v_T = v_L = 0$.

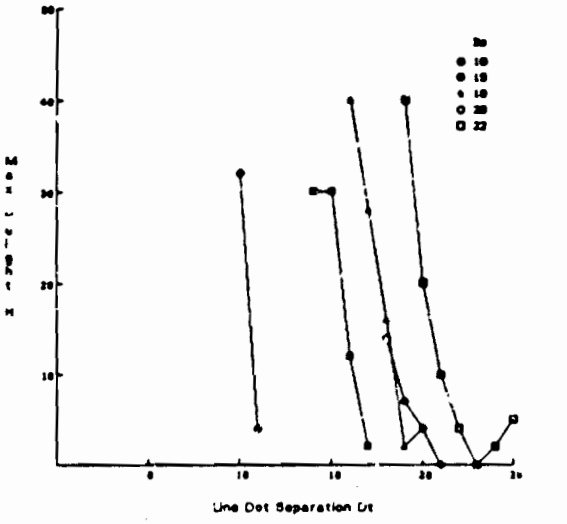
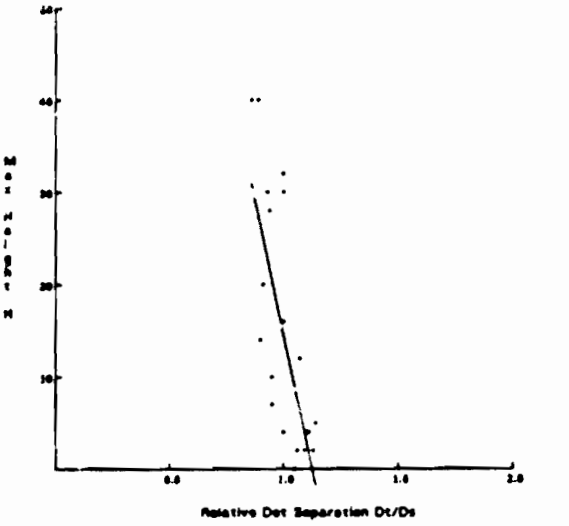


Figure 10. Maximum H as a function of dot separation d_t. Curves have constant d_s. (The upswing at the end of two of the curves is probably due to noise in the data.)



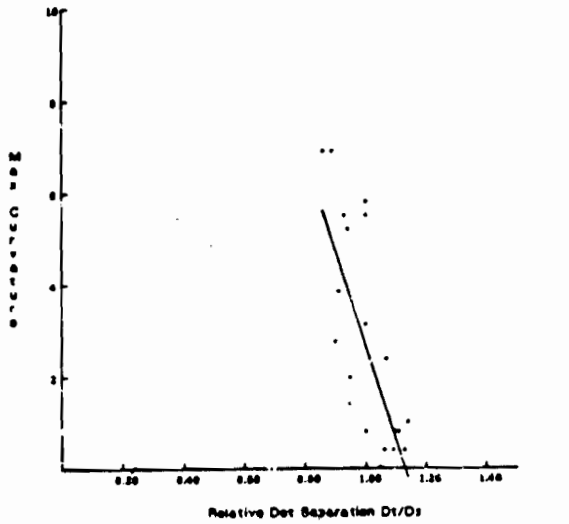


Figure 11. (a) Maximum H as a function of relative dot separation d_t/d_s. Fitted line has a slope of -116 and a correlation of -0.72. (b) Maximum curvature κ as a function of d_t/d_s. (The curvature values plotted are 1000 times the computed value, for ease of reading.) This looks very similar to (a) because curvature as a function of H is nearly linear in the range $(0, 50)$.

On the other hand, jaggedness, or deviation from a straight line, (as measured by v_T) is an important determiner of performance; as the amount of transverse variation increases, performance drops off. The results of these experiments show that the amount of jaggedness we can tolerate falls off linearly with the separation ratio. Performance at the detection task (for straight lines) seems to depend only on the ratio of the spacing between the dots in the target to the average spacing between dots in the surround. This suggests that the surround is taken into account by some sort of local difference mechanism, which has yet to be elucidated.

Curvature plays an important role in detectability as well. These experiments confirm the intuition that curved dotted lines should be more difficult to perceive than straight ones, that is, the dots in the line need to

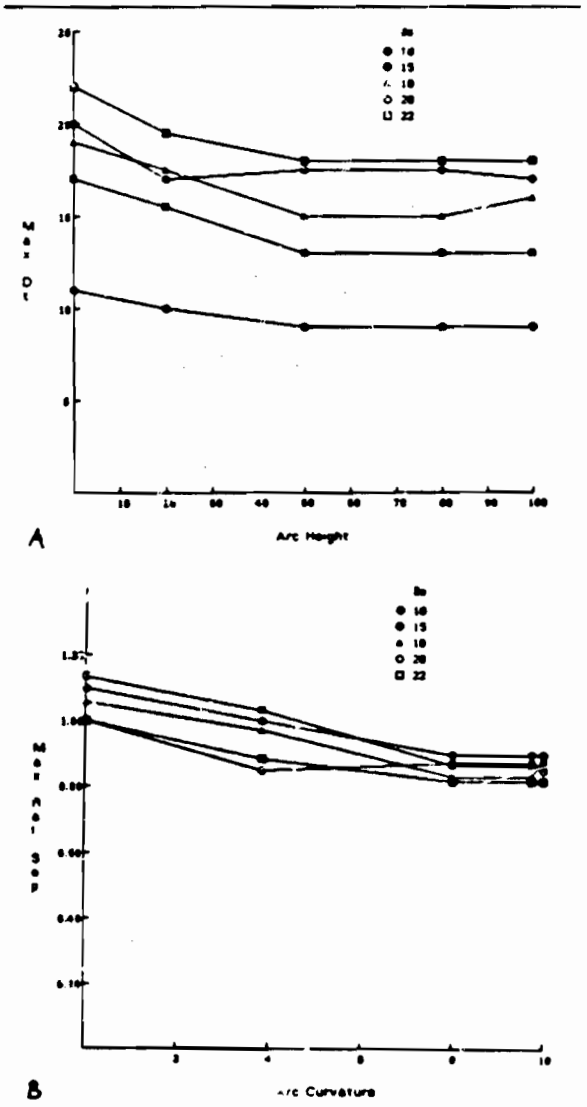


Figure 12. (a) Maximum separation of dots in arc d_s as a function of arc height H. Curves have constant d_o. (b) Maximum relative separation d_s/d_o of dots in arc as a function of arc curvature. Note that the curve levels off. (Curvature shown is again actually 1000 times actual curvature.)

be closer together relative to the surround. However, the experiments show that the effects of curvature level off after a point, and that further curvature increases do not necessitate decreasing the dot distance in the arc.

Possible Mechanisms

I shall now speculate on mechanisms that might be able to produce behavior similar to that found in these experiments, both in computers and in biological vision systems. I will present an outline of a mechanism that shows promise of solving the problem of detecting the kinds of targets considered here, as well as two other related, and important, problems in early vision.

Two related problems

Consider the random-dot display in Figure 13. The impression of bilateral symmetry is immediate, and is in fact pre-attentive (see Barlow and Reeves (1979) and Barlow (1980)). Detecting symmetry of shapes and patterns is a problem of obvious importance for animals, and is one of practical importance for computer vision systems as well. The problem of finding parallel lines (Figure 14(a)) and parallel curves (Figure 14(b)) is closely related. A single mechanism should be able to find all these kinds of symmetry.

Now look at the image in Figure 15. This picture creates a strong impression of two different textures. The two sides have identical dot densities but a different rule for placing the dots. The dots in one texture are "more random" than those in the other texture. The dots on each side were in fact generated by the algorithm described early in this paper (see *Details*), with v_r different for the two sides. I tested how well subjects can discriminate two textures with regularity differences such as this, and the results show that, as expected, when the variations in regularity are quite different, the textures are easy to discriminate, and for those that are more similar, the discrimination becomes more difficult. The question is, what kind of mechanism could detect texture differences such as this?

Some related results

Caelli (1981; see also Caelli *et al*, 1978) has noted that human beings have the ability to segment textures that differ in their θ-variability, that is, textures composed of line segments whose orientations fall in a larger range in one region than in another. For example, in Figure 16, the left side has line segments whose orientations are $60 \pm 10°$ while those on the right side have orientations of $60 \pm 30°$. Thus they have the same average orientation but a different variability or range. (Caelli's experiments used dipoles, or dot pairs. The results are similar.) We can also segment regions of small θ-variability from regions of random orientation, as shown in Figure 17.

In fact, we can segment regions of just two, or even three, fixed orientations from a region of random orientations, as shown in Figure 18. This is in contradiction with the results of Riley (1981), who claimed that two fixed orientations cannot be segmented from random orientations. However, his demonstration did not allow a large enough area on which to base our comparisons. That is, there was not enough information in the picture to enable us to make a segmentation decision. I have performed

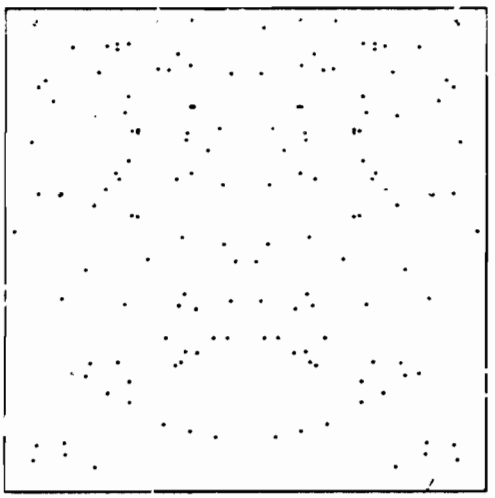


Figure 13. A bilaterally symmetric pattern.

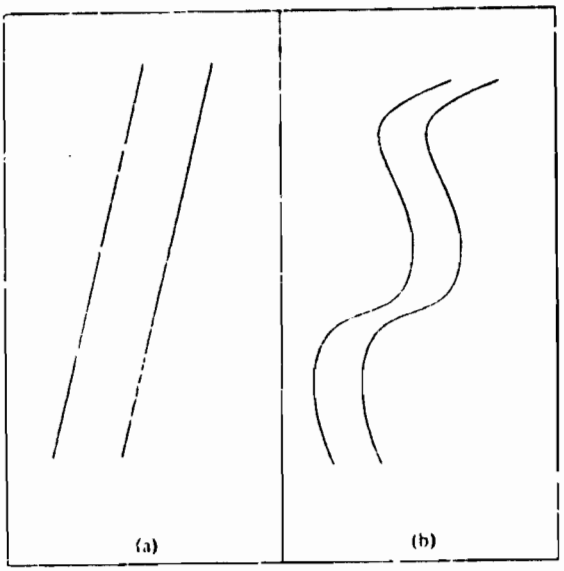


Figure 14. (a) Parallel lines; (b); parallel curves.

some informal experiments that show that as the width of the two-orientation region decreases, the segmentation becomes more difficult. Riley's demonstration showed that the segmentation is very difficult when the width is small.

Proposal

Suppose, then, that virtual line segments are constructed in the image between each dot and some small number of its nearest neighbors. Let the "weight" of each segment be inversely related (in some as-yet unspecified way) to the distance between the dots at its ends. Then at each dot location in the image there will be a number of oriented virtual line segments, some of them weighted

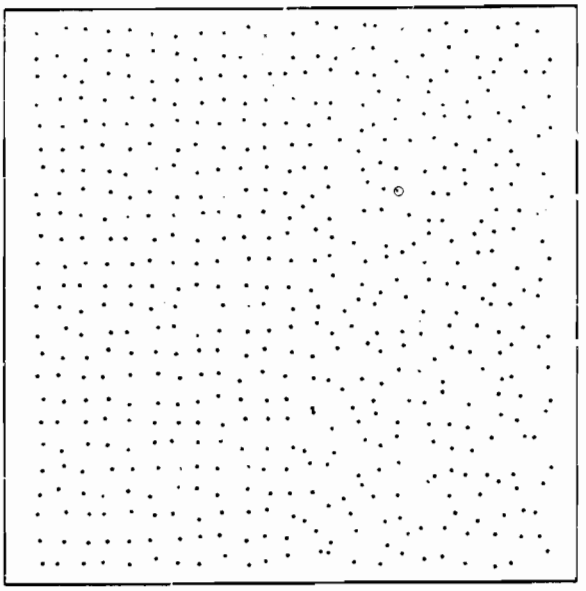


Figure 15. The texture difference in this image is brought about by a difference in the regularity of placement of the dots on the two sides. On the left, the variation $v = 0.4$; on the right it is $v = 0.8$. The right side looks more "random."

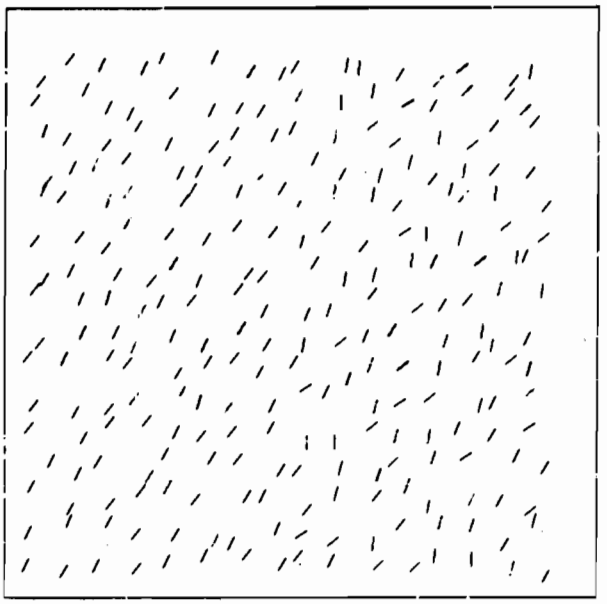


Figure 16. A difference in orientation variability creates a texture difference. The mean orientation is the same on both sides (60°), but on the left it varies by ±10° while on the right it varies by ±30°.

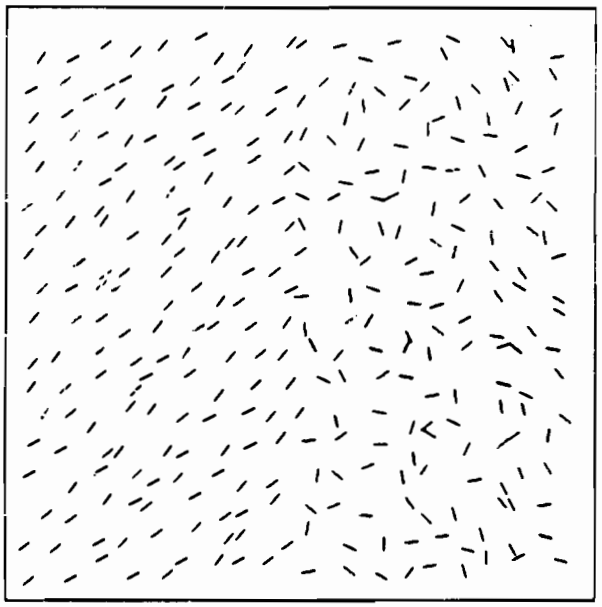


Figure 17. Line segments of a single orientation, with small variability, can be segmented from random orientations. The segments on the left are oriented at $40 \pm 15°$; on the right they are randomly oriented.

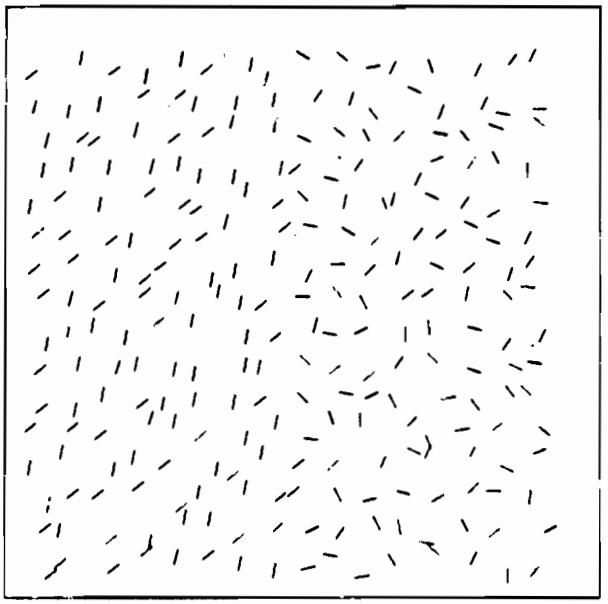


Figure 18. A region composed of lines with just two orientations can be segmented from a region composed of randomly-oriented lines. On the left the orientations are 40 and 80 degrees.

more than the others. Suppose now that the image consists of a linear string of dots, with no surround. Then all the virtual line segments will be aligned with this linear feature. If we now surround the dotted line with random dots (of sufficient sparseness), the predominant orientations of the virtual line segments will still be along the line. It is clear that as the distance between the dots in the surround gets smaller and approaches the distance between the dots in the line, the orientations of the virtual lines become more and more random since those orientations are influenced to a greater degree by nearby random dots. If we could segment regions of small variability in orientation from regions of large variability, we could perhaps detect the target.

To see whether this approach is useful, I performed some simulations of the process of forming virtual lines on images of dotted lines embedded in a random surround. The weighting function I used for these simulations was $w(r) = e^{-\alpha r}$, as suggested by Caelli *et al* (1978). The value of α was 0.05 (pixels^{-1}). Two ways to process the results of this virtual-line finding process come to mind: the first is to use all the virtual line segments that are generated, and the second is to form a weighted average of the orientations. In the second approach, it is necessary to combine the orientations of line segments with orientations θ and $\theta + 180$; this can be done by the trick (see Caelli *et al* (1978) and Kass and Witkin (1985)) of first multiplying the angle by two, then adding the resulting vectors, and finally dividing the angle of the vector sum by two to obtain the resultant virtual line. I tried both of these approaches, and present some results in Figures 19and 20. Note that when the dots in the line are close together compared with those in the surround, as in Figure 19, the predominant orientation of virtual segments in the line is along the line, while in the surround it is mostly random. Compare this to Figure 20, where d_t is about the same as d_s; the experiments reported earlier show that it is difficult for people to detect the line, and the orientations indeed appear to be mostly random along the line.

How could this field of oriented line segments be used to detect linear features (or arcs, for that matter)? A mechanism that separates regions of small variation from regions of large variation would be required. The experiments of Caelli noted above show that people can in fact perform this task. To see how this might be performed computationally, consider several operators corresponding to each dot location in the image, each operator aligned with a different orientation (e.g., an operator aligned every 15°). Each operator responds to dot pairs in a small range of orientations, and its output varies inversely with the distance between the dots. The reader may visualize this situation as a set of planes aligned with the image, with all the operators in each plane aligned in the same direction. (This bears a certain resemblance, not coincidentally, with the well-known orientation columns that Hubel and Weisel discovered in the visual cortex of mammalian brains; see, for example, Hubel and Weisel (1968), Hubel *et al* (1978), and Hubel (1979).)

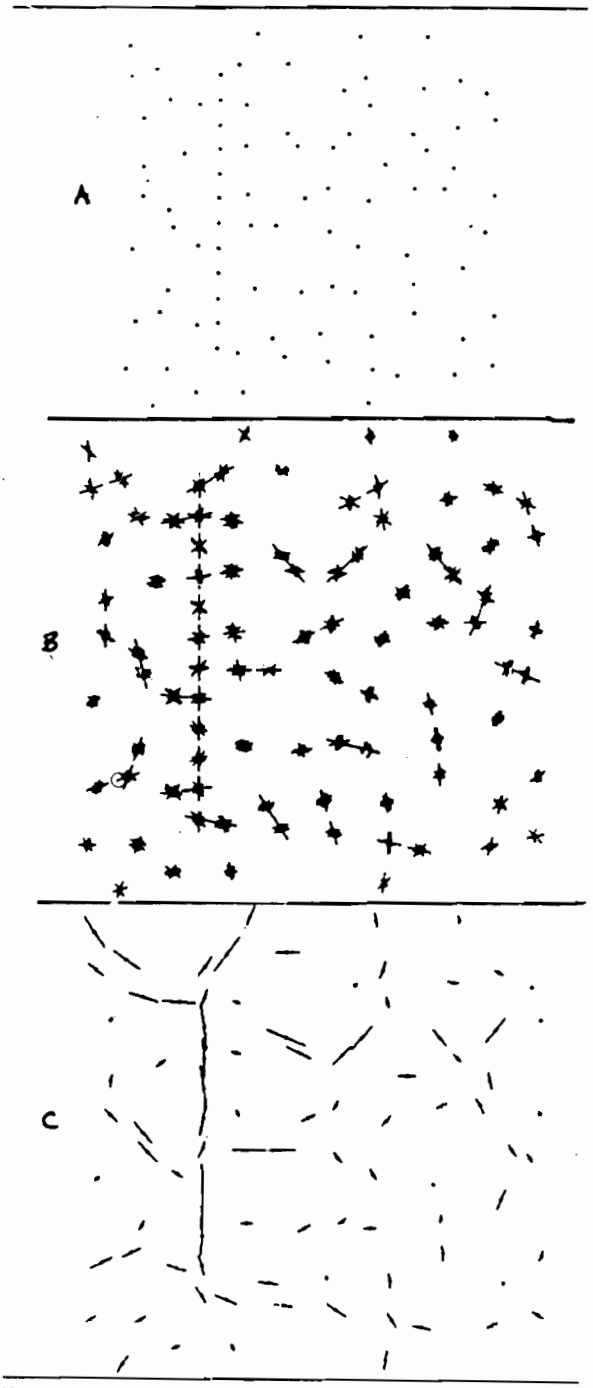


Figure 19. Simulation of the virtual-line finding process. (a) The original image. Parameters are: $d_t = 12$, $d_e = 20$. (b) Oriented virtual lines are formed at each dot location between that dot and its 10 nearest neighbors. The weight of each virtual line is represented here by its length, and is inversely related to the distance between the dots from which it was formed. (c) The weighted average of all the virtual line segments is displayed for each dot position. The predominant orientation of segments along the line is with the line.

Now, a set of randomly-oriented lines in the image will produce output from a corresponding random set of operators. But a region of constant, or slightly varying, orientations surrounded by a region of random orientations will produce output in a region of operators on one plane, with random output in the other planes. The problem then reduces to one of finding a region of differing density of features, that is, finding a cluster. This problem is somewhat similar to the problem of finding intensity edges in images, except that the data points are much more sparse. I am currently designing an algorithm to perform this clustering process.

So it appears that the problem of finding linear strings of dots in noisy surrounds can be solved by the method of virtual line orientations. What about the other two related problems posed above? Let us consider the problem of discriminating two textures that differ in the regularity of dot placement. If virtual lines are formed between nearby dots on each side, then the orientations of the dots on the more regular side will tend to line up in two dominant directions (in this case, with the horisontal and vertical axes); those on the other side will be more random. There will be two predominant orientations in the more regular side, and none in the random side. Two regions in two orientation planes will produce output, while the rest will be random. Recall the demonstration in Figure 18 that showed that it is possible to segment a texture composed of lines at two orientations from one with lines at random orientations. The orientation-column approach outlined above should be sufficient to perform this task as well. Thus, this mechanism seems to be sufficient to detect regularity differences such as the one demonstrated here.

The mechanism may also be able to detect symmetry in patterns. At first glance, one might expect that there would be a predominant orientation across the axis of symmetry, corresponding to symmetric pairs of dots. However, the simple one-level mechanism presented here is not sufficient to detect most instances of symmetry. For example, in Figure 13, if we apply the virtual-line forming process, only the dots near the axis of symmetry are close enough to their symmetric mates for a virtual line of significant weight to be formed. In most places, the virtual orientations are random and cannot help us find the symmetry. However, I believe that the mechanism can still be used. If higher-level features, such as short line segments and clusters, are formed on each side, and the original dots removed, then the algorithm can be applied to the positions of these features. If appropriate features are chosen so that the density of features is smaller at this higher level, then the nearest neighbors of features will include their symmetric mates. There will then be a peak in orientation of the virtual line segments across the axis of symmetry; this should enable the detection of the symmetry.

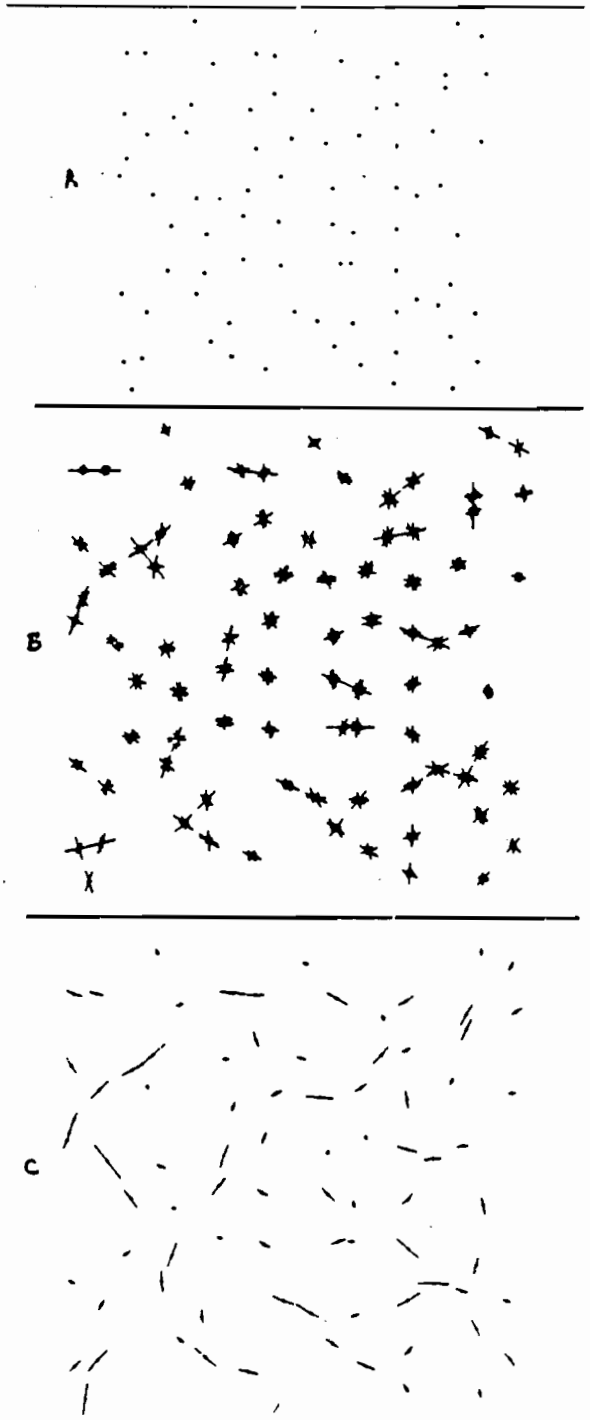


Figure 20. Another simulation of the virtual line-finding process, this time with parameters $d_t = d_s = 20$. The virtual orientations are essentially random.

Conclusions

I have presented the results of several psychophysical experiments in preattentive vision. The experiments sought to determine the important parameters that affect the detectability of dotted lines and dotted arcs when they are embedded in noisy backgrounds. Results indicate that the length of the target is not important, and neither is the regularity of spacing of the dots along the target's axis. The amount of deviation from a straight line does affect performance, as we intuitively expect. The amount of curvature in an arc also affects performance. The results indicate that performance seems to depend on the ratio of the separation of the dots in the target to the separation of those in the noisy surround, rather than upon either independently. That is, detection does not seem to be diameter-limited but rather complexity-limited, in the sense of Binford (1983).

I made some speculations on a mechanism that could perform this task of detecting dotted targets in a noisy surround. I claimed that this same mechanism might also be able to perform the task of finding symmetry and parallelism in images, as well as segmenting textures that differ in the spatial regularity of their elements. This work is clearly in its preliminary stages, but shows promise for future work in machine detection of patterns such as those considered here.

Acknowledgements

I would like to thank my advisor Tom Binford for his help and encouragement, and Brian Wandell for his useful suggestions. This research was supported by ARPA contract N00039-84-C-0211.

References

Barlow, H.B. and B.C. Reeves (1979). "The versatility and absolute efficiency of detecting mirror symmetry in random dot displays." *Vision Research*, Vol. 19, 783–793.

Barlow, H.B. (1980). "The absolute efficiency of perceptual decisions," *Phil. Trans. R. Soc. Lond. B* 290, 71–82.

Binford, T.O. (1983). "Figure/ground: Segmentation and aggregation," in Braddick, O.J. and A.C. Sleigh (eds.), *Physical and Biological Processing of Images*. New York: Springer-Verlag.

Caelli, Terry (1981). *Visual Perception: Theory and practice*. Oxford, England: Pergamon Press.

Caelli, T.M., G.A.N. Preston and E.R. Howell (1978). "Implications of spatial summation models for processes of contour perception: A geometric perspective." *Vision Res.* Vol. 13, 723–734.

Hubel, D.H. (1979). "The visual cortex of the brain," *Scientific American*, Nov. 1979.

Hubel, D.H. and T.N. Wiesel (1968). "Receptive fields and the functional architecture of monkey striate cortex," *J. Physiol. (Lond.) 195*, 215–243.

Hubel, D.H., T.N. Wiesel, and M.P. Stryker. (1978) "Anatomical demonstration of orientation columns in Macaque monkey." *J. Comp. Neur.*, 177, 361-380.

Kass, M. and A. Witkin (1985). "Analyzing oriented patterns". Proc. IJCAI-85.

Lowe, D. (1984). "Perceptual organization and visual recognition." Ph.D. Thesis, Computer Science Dept., Stanford University.

Lowe, D. (1985). "Visual Recognition from Spatial Correspondence and Perceptual Organization." Proc. IJCAI-85.

Neisser, U. (1967). *Cognitive Psychology*. New York: Appleton-Century-Crofts.

Riley, Michael D. (1981). "The representation of image texture." MIT AI Lab Memo TR-649, September, 1981. Master's Thesis.

Triesman, A. (1985). "Preattentive processing in vision," *Comp. Vis., Graphics, and Image Proc.*, 1-22.

Witkin, Andrew P. and Jay M. Tenenbaum (1982). "On the role of structure in vision." In Jacob Beck, B. Hope, and A. Rosenfeld (eds.), *Human and Machine Vision*. New York: Academic Press.

Zucker, Steven W. (1982). "Computational and psychophysical experiments in grouping: Early orientation selection." In Jacob Beck, B. Hope, and A. Rosenfeld (eds.), *Human and Machine Vision*. New York: Academic Press.

Zucker, S.W. (1983) "Cooperative grouping and early orientation selection," in Braddick, O.J. and A.C. Sleigh (eds.), *Physical and Biological Processing of Images*. New York: Springer-Verlag.

ONE-EYED STEREO: A GENERAL APPROACH TO MODELING 3-D SCENE GEOMETRY

Thomas M. Strat and Martin A. Fischler *

Artificial Intelligence Center
SRI International, Menlo Park, California

October 23 1985

Abstract

A single two-dimensional image is an ambiguous representation of the three-dimensional world - many different scenes could have produced the same image—yet the human visual system is extremely successful at recovering a qualitatively correct depth model from this type of representation. Workers in the field of computational vision have devised a number of distinct schemes that attempt to emulate this human capability; these schemes are collectively known as "shape from ...." methods (*e.g.*, shape from shading, shape from texture, or shape from contour). In this paper we contend that the distinct assumptions made in each of these schemes must be tantamount to providing a second (virtual) image of the original scene, and that any one of these approaches can be translated into a conventional stereo formalism. In particular, we show that it is frequently possible to structure the problem as one of recovering depth from a stereo pair consisting of the supplied perspective image (the *original* image) and an hypothesized orthographic image (the *virtual* image). We present a new algorithm of the form required to accomplish this type of stereo reconstruction task.

1 Introduction

The recovery of 3-D scene geometry from one or more images, which we will call the scene-modeling problem (SMP), has solutions that appear to follow one of three distinct paradigms: stereo; optic flow; and shading, texture, and contour.

In the stereo paradigm, corresponding world/scene points in two images are used, together with the relative geometry of the two cameras (which we can also derive), we can use simple trigonometry to determine the locations of the matched points [1].

In the optic flow paradigm, a sequence of time-varying images is used to compute the image velocities of the various points. If the camera's motion is known, or can be derived, we can again use simple trigonometry to convert velocity measurements in the image to depths in the scene.

In the shape from: shading, texture, and contour (SSTC) paradigm, we must either know, or make some assumptions about the nature of the scene, the illumination, and the imaging geometry. Brady's 1981 volume on computer vision [2] contains an excellent collection of papers, many of which address the problem of how to recover depth from the shading, texture, and contour information visible in a single image. Two distinct computational approaches have been employed in the SSTC paradigm: (1) integration of partial differential equations describing the relation of shading in an image to surface geometry in a scene, and (2) back-projection of planar image facets to undo the distortion in an image attribute (*e.g.*, the orientation) induced by the imaging process on an assumed scene property (*e.g.*, uniform distribution of edge orientations).

Our purpose in this paper is to provide a unifying framework for the scene modeling problem, and to present a new computational approach to recovering scene geometry from the shading, texture, and contour information in a single image. Our contribution is based on the following observation: regardless of the assumptions employed in the SSTC paradigm, if a 3-D scene model has been derived successfully, it will generally be possible to establish a large number of correspondences between image and scene (model) points. From these correspondences we can compute a collineation matrix [11], and then extract the imaging geometry from it [1, 39]. We can now construct a second image of the scene as viewed by the camera from some arbitrary location in space. It is thus obvious that any technique that is competent to solve the SMP must either be provided with at least two images or make assumptions that are equivalent to providing a second image. We can unify the various approaches to the SMP by converting their respective assumptions and auxiliary information into the implied second image and employing the stereo paradigm to recover depth. In the case of the SSTC paradigm, our approach amounts to "one-eyed stereo."

2 Shape from One-Eyed Stereo

Most people viewing Figure 1 get a strong impression of depth. We can recover an equivalent depth model by assuming that we are viewing a projection of a uniform grid and employing the computational procedure to be described. In the remainder of this paper we will show how some simple modifications and variations of the uniform grid, as the implied second image, allow us to recover depth from shading, texture, and contour.

The one-eyed stereo paradigm can be described as a five-step process, as outlined in the paragraphs below. Some scenes with special surface markings or image-formation processes must be analyzed by variants of the algorithm described, but the general approach remains the same.

The work reported herein was supported by the Defense Advanced Research Projects Agency under Contract MDA903-83-C-0027.

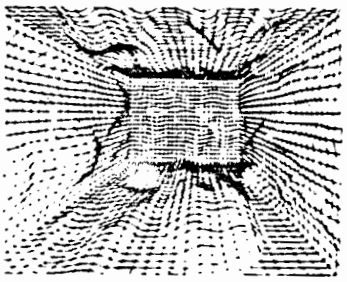


Figure 1: Wire Room

2.1 Partition the Image

As with all approaches to the SMP, the image must be segmented into regions prior to the application of a particular algorithm. Before the one-eyed stereo computation can be employed, the segmentation process must delineate regions that are individually in conformance with a single model of image formation. The computation can then be carried out independently in each region, and the results fitted together.

2.2 Select a Model

For each region identified by the partitioning process, we must decide upon the underlying model of image formation that explains that portion of the image. Surface reflectance functions and texture patterns are examples of such models. Partitioning of the image and selection of the appropriate models are difficult tasks that are not addressed in this paper. Witkin and Kass [23] are exploring a new class of techniques that promises some eventual answers to these questions. Generally, it will be impossible to recover depth whenever a single model cannot be associated with a region. Similarly, inaccurate or incorrect results can be expected if the partitioning or modeling is performed incorrectly.

2.3 Generate the Virtual Image

The key to one-eyed stereo is using the model of image formation to fabricate a second (virtual) image of the scene. The idea is that the model often allows one to construct an image that is independent of the actual shape of the imaged surface. This allows the virtual image to be depicted solely from knowledge of the model without making use of the original image. For example, the markings on the surface of Figure 2(a) could have arisen from projection of a uniform grid upon the surface. For all images that fit this model, we can use a uniform grid as the virtual image. As a rule, the orientation, position, and scale of this grid will be unknown, however, we will show how this information can be recovered from the original image. Other models give rise to other forms of virtual images.

2.4 Determine Correspondences

Before applying stereo techniques to calculate depths, we must first establish correspondences between points in the real image and the virtual image. When dealing with textures, the process is typified by counting texels in each image from a chosen starting point. With shaded images, the general approach

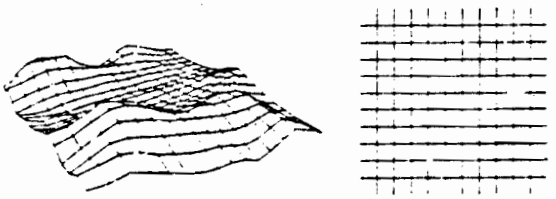


Figure 2: (a) A projected texture (b) Its virtual image

is to integrate intensities. Several variants of the method for establishing correspondences are described in the next section. The difficulty of the procedure, it should be noted, will depend on the nature of the model.

2.5 Compute Depths Using Stereo

With two images and a number of point-to-point correspondences in hand, the techniques of binocular stereo are immediately applicable. At this point, the problem has been reduced to computing the relative camera models between the two images and using that information to compute depths by triangulation. The fact that the virtual image will normally be an orthographic projection required reformulation of existing algorithms for performing this computation. The appendix describes a new algorithm that computes the relative camera model and reconstructs the 3-D scene from eight point correspondences between a perspective and an orthographic image.

The problem of recovering scene and imaging geometry from two or more images has been addressed by workers not only in binocular stereo, but also in monocular perception of motion in which the two projections are separated in time as well as space. Various approaches have been employed to derive equations for the 3-D coordinates and motion parameters; these equations are generally solved by iterative techniques [5] [8] [13] [14]. Ullman [21] presents a solution for recovering 3-D shape from three orthographic projections with established correspondences among at least four points. His "polar equation" allows computation of shape when the motion of the scene is restricted to rotation about the vertical axis with arbitrary translation. Nagel and Neumann [10] have devised a compact system of three nonlinear equations for the unrestricted problem when five point correspondences between the two perspective images are known. More recently, Huang [20] and Longuet-Higgins [9] have independently derived methods requiring only that a set of eight simultaneous linear equations be solved when eight point correspondences between two perspective images are known. In our formulation we are faced with a stereo problem involving a perspective and an orthographic image, while the aforementioned references are indeed germane, none provides a solution to this particular problem.

The derivation described in the appendix was inspired by the formulation of Longuet-Higgins for perspective images. When either image nears orthography, Longuet-Higgins's method becomes unstable, it is undefined if either image is truly orthographic. Moreover, his approach requires knowledge of the focal length and principal point in each image while our method was derived specifically for one orthographic and one perspective image whose internal imaging parameters may not be fully known.

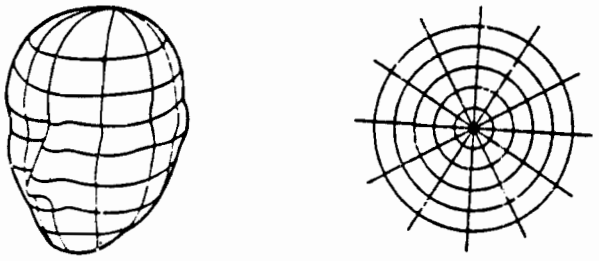


Figure 3: (a) The original image (b) The virtual image

3 Variations on the Theme

In this section we illustrate how our approach is used with several models of texture, shading, and contour. Where these models do not match given scene characteristics, they may require additional modification. However, a qualitatively correct answer might still be obtainable by applying one of the specific models we discuss below to a situation that appears to be inappropriate, or to an image in which the validity of the assumptions cannot be established.

3.1 Shape from Texture

Surface shapes are often communicated to humans graphically by drawings like Figure 2(a). Such illustrations can also be interpreted by one-eyed stereo. In this case, there is no need to partition the image; the underlying model of the entire scene consists of the intersections of lines distributed in the form of a square grid. When viewed directly from above at an infinite distance, the surface would appear as shown in the virtual image of Figure 2(b) regardless of the shape of the surface. This virtual image can be construed as an orthographic projection of the object surface from a particular, but unknown, viewing direction. Correspondences between the original and virtual images are easily established if there are no occlusions in the original image. Select any intersection in the original image to be the reference point and pair it with any intersection in the virtual image. A second corresponding pair can be found by moving to an adjacent intersection in both images. Additional pairs are found in the same manner, being careful to correlate the motions in each image consistently in both directions. When occlusions are present, it may still be possible to obtain correspondences for all visible junctions by following a nonoccluded path around the occlusion (such as the hill in the foreground of Figure 2(a)). If no such path can be found, the shape of each isolated region can still be computed, but there will be no way to relate the distances without further information. Other techniques used to represent images of 3-D shapes graphically may require other virtual images. Figure 3(a), for example, would imply a virtual image as shown in Figure 3(b). Methods for recognizing which model to apply are needed, but are not discussed here.

Once correspondences have been determined, we can use the algorithm given in the appendix to recover depth. We have presumably one perspective image and one orthographic image whose scale and origin are still unknown. The depths to be recovered will be scaled according to the scale chosen for the virtual image[1]. The choice of origin for the orthographic image is arbitrary, and will lead to the same solution regardless of the point chosen. The appendix shows how to compute both the orientation and the displacement of the orthographic coordinate system, relative to the perspective imaging system. 3-D coordinates of each matched point are then easily computed by means of back-projection. A unique solution will be obtained whenever the piercing point or focal length of the perspective image is known. A minimum of eight pairs of matched points is required to obtain a solution; depths can be computed for all matched points.

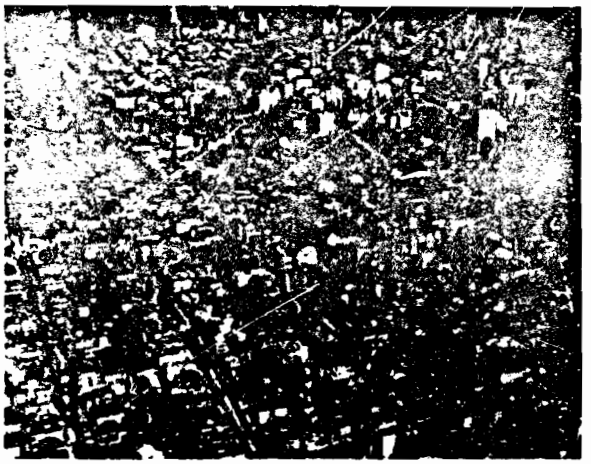


Figure 4: The streets in this scene resemble a projected texture. [3]

There exists a growing literature on methods to recover shape from natural textures [7][12][18][22]. We will now show how the constraints imposed by one type of natural texture can be exploited to obtain similar results by using one-eyed stereo.

Consider the pattern of streets in Figure 4. If this city were viewed from an airplane directly overhead at high altitude, the streets would form a regular grid not unlike the one used as the virtual image in Figure 2. There are many other scene attributes that satisfy this same model. The houses in Figure 5 would appear to be distributed in a uniform grid if viewed from directly overhead. In an apple orchard growing on a hillside, the trees would be planted in rows that are evenly spaced when measured horizontally; the vineyard in Figure 6 exhibits this property.

Ignoring the nontrivial tasks of partitioning these images into isotextural regions, verifying that they satisfy the model, and identifying individual texels, it can be seen how these images can be interpreted with the same techniques as were described in the previous section. The virtual image in each case will be a rectangular grid that can be considered as an orthographic view from an unknown orientation. Correspondences can be established by counting street intersections, rooftops, or grape vines. As before, one can solve for the relative camera model and compute depths of matched points. Obviously, for the situations discussed here, we must be satisfied with a qualitatively

[1] Recall that the original image does not contain the information necessary to recover the absolute size of the scene.

Figure 6: These grapevines exhibit a regular texture. [3]

correct interpretation—not only because of the difficulty of locating individual texels reliably and accurately, but also in view of the numerical instabilities arising from the underlying nonlinear transformation.

3.2 Shape from Shading

For our purposes, surface shading can be considered the limiting case of a locally uniform texture distribution (as the texels approach infinitesimal dimensions). To compute correspondences, we need to integrate image intensities appropriately in place of counting lines, since the image intensities can be seen to be related to the density of lines projected on the surface. The feasibility of this procedure depends on the reflectance function of the surface.

What types of material possess the special property that allows their images to be treated like the limiting case of the projected textures of the previous section? The integral of intensity in an image region has to be proportional to the number of texels that would be projected in that region. if the angles i and e are defined as depicted in Figure 7, it can be seen that the number of texels projected onto a surface patch will be proportional to $\cos i$, the cosine of the incident angle. At the same time, the surface patch (as seen from the viewpoint) will be foreshortened by $\cos e$, the cosine of the emittance angle. Thus, the integral

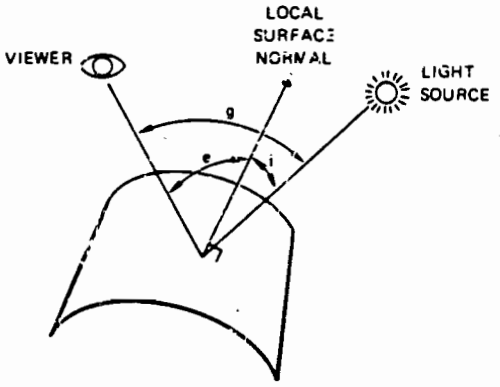


Figure 7: The geometry of surface illumination

of reflected light intensity over a region will be proportional to the flux of the light striking the surface if the intensity of the reflected light at any point is proportional to $\cos i / \cos e$. Horn [6] has pointed out that, when viewed from great distances, the material in the maria of the moon and other rocky, dusty objects exhibit a reflectance function that allows recovery of the ratio $\cos i / \cos e$ from the imaged intensities. This surface property has made possible unusually simple algorithms for computing shape-from-shading, so it is not surprising that it submits easily to one-eyed stereo as well.

To interpret this type of shading, we can construct a virtual image whose direction of view is the lighting direction (i.e., taken from a "virtual camera" located at the light source). When the original shaded image is orthographic, we consider a family of parallel lines in which each line lies in a plane that includes both the light source and the (distant) viewpoint. When viewed from the light source, the image of the surface corresponding to these lines will also be a set of parallel lines regardless of the shape of the surface. These parallel lines constitute the virtual image. We will use the image intensities to refine these line-to-line correspondences to point-to-point correspondences. Figure 8 shows the geometry for an individual line in the family. A little trigonometry shows that

$$\Delta s' = \frac{\cos i}{\cos e} \Delta s \quad , \tag{1}$$

where Δs is a distance along the line in the real image and $\Delta s'$ is the corresponding distance along the corresponding line in the virtual image. Integrating this equation produces the following expression, which defines the point correspondences in the two images along the given line.

$$s' = s'_0 + \int_0^s \frac{\cos i}{\cos e} ds \tag{2}$$

To use this equation we must first compute $\frac{\cos i}{\cos e}$ from the intensity value at each point along the line. This will, of course, be possible only when the reflectance function is constant for constant $\frac{\cos i}{\cos e}$. Next we choose a starting point in the shaded image and begin integrating intensities according to Equation (2). For any value of s, the corresponding virtual image point is along a straight line at a distance s' from the virtual reference point. With these point-to-point correspondences in hand, it

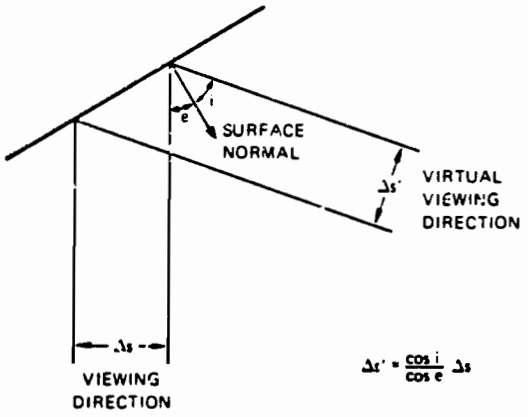


Figure 8: The geometry along a line in the direction of the light source

is a simple matter of triangulation to find the 3-D coordinates of the surface points, given that we know the direction to the light source. We can explore the remainder of the surface by repeating the process for each of the successive parallel lines in the image. Adjacent profiles still remain unrelated to each other, since their individual scale factors have not yet been ascertained. Knowledge of the actual depth of one point along each profile provides the necessary additional information to complete the reconstruction. It is important to note that our assumptions and initial conditions are those used by Horn; the fact that he was able to obtain a solution under these conditions assured the existence of a suitable virtual image for the one-eyed stereo paradigm.

For shaded perspective images, we must integrate along a family of straight lines that radiate from the point in the image that corresponds to the location of the light source. This ensures that the image line will be in a plane containing both the viewer and the light source, and that the virtual image of each line will also be a straight line. The integration becomes a bit more complex than shown in Equation 2 because the nonlinear effects of perspective imaging must be accommodated. Nevertheless, it remains possible to establish point-to-point correspondences between images and to reconstruct the surface along each line.

3.3 Shape from Contour

It is sometimes possible to extract a line drawing, such as the one shown in Figure 9, from scene textures. Parallel streets like those encountered in Figure 4 give rise to a virtual image consisting of parallel lines when the cross streets cannot be located; terraced hills also produce a virtual image of parallel lines. Correspondences between real and virtual image lines can be found by counting adjacent lines from an arbitrary starting point. This matches a virtual image line with each point in the real image. Point-to-line correspondences are not sufficient to enable the stereo computation of the appendix to be used for reconstruction of the surface. Knowledge of the relative orientation between the two images (equivalent to knowing the orientation of the camera that produced the real image relative to the parallel lines in the scene) provides an adequate constraint; the surface can then be reconstructed uniquely through back-projection. Without knowledge of the relative orientation of the virtual image, heuristics must be employed that relate points on adjacent contours so that a regular grid can be used as the virtual image. The human visual system is normally able to interpret images like Figure 9 unambiguously although just what assumptions are being made remains unclear. Further study of this phenomenon may make it possible to extract models that are especially suited to the employment of one-eyed stereo on this type of image without requiring prior knowledge of the virtual orientation.

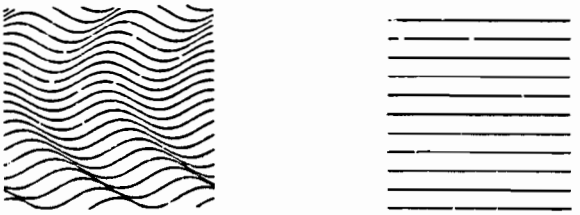


Figure 9: (a) An image of contours (b) Its virtual image

3.4 Distorted Textures and Unfriendly Shading

We have already noted that image shading can be viewed as a limiting (and, for our purposes, a degenerate) result of closely spaced texture elements. To recover depth from shading, we must use integration instead of the process of counting the texture elements that define the locations of the "grid lines" of our virtual image. The integration process depends on the existence of a "friendly" reflectance function and an imaging geometry that allows us to convert distance along a line in the actual image to a corresponding distance along a line in the virtual image.

The recovery of lunar topography from a single shaded image [6], as discussed in Section 3.2, is one of the few instances in which "shape from shading" is known to be possible without a significant amount of additional knowledge about the scene. Nevertheless, even here we are required to know the actual reflectance function, the location of the [point] source of illumination, and the depths along a curve on the object surface, and be dealing with a portion of the surface that has constant albedo. Furthermore, the reflectance function has to have just the property we require to replace direct counting, i.e., the reflectance function has to compensate exactly for the "foreshortening" of distance due to viewing points on the object surface from any angle. Most of the commonly encountered reflectance functions, such as Lambertian reflectance, do not possess this friendly property, and it is not clear to what extent it is possible to recover depth from shading in such cases (e.g., see Pentland [12] and Smith [15]). Additional assumptions will probably be necessary and the qualitative nature of the recovery will be more pronounced. Just as in the case in which a complex function can be evaluated by making a local linear approximation and iterating the resulting solution, so it may be possible to deal with unfriendly, or even unknown, reflectance functions by assuming

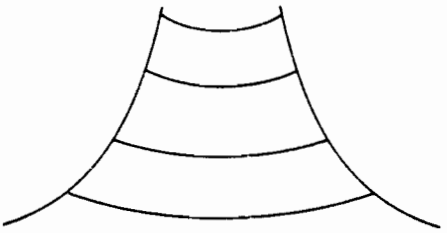


Figure 10: This simple drawing has two reasonable interpretations. It is seen as curved roller-coaster tracks if the lines are assumed to be the projection of a rectangular grid, or as a volcano when the lines are assumed to be the projection of a circular grid.

that they are friendly in the vicinity of some point, solving directly for local shape by using the algorithm applicable to the friendly case, and then extending the solution to adjacent regions. We are currently investigating this approach.

The uniform rectangular grid and the polar grid that we used as virtual images to illustrate our approach to one-eyed stereo are effective in a large number of cases because there are processes operating in the real world that produce corresponding textures (i.e., gridlike textures that appear to be orthographically projected onto the surfaces of the scene). However, there are also textures that produce similar-appearing images, but are due to different underlying processes. For example, a uniform gridlike texture might have been created on a flat piece of terrain that is subsequently subjected to geologic deformation—in this case the virtual image (or the recovery algorithm) needed to recover depth must be different from the projective case. We have already indicated the problem of choosing the appropriate model for the virtual image and, as noted above, image appearance alone is probably insufficient for making this determination—some semantic knowledge about the scene is undoubtedly essential. Figure 10 shows an example in which two completely different, yet equally believable, interpretations of scene structure result, depending on whether we use the rectangular grid model or the polar grid model.

4 Experimental Results

The stereo reconstruction algorithm described in the appendix has been programmed and successfully tested on both real and synthetic imagery. Given a sparse set of image points and their correspondence in a virtual image, a qualitative description of the imaged surface can be obtained.

Synthetic images were created from surfaces painted with computer-generated graphic textures. Figure 11(a) shows a synthetic image constructed from a section of a digital terrain model (DTM). The intersections of every twentieth grid line constitute the set of 36 image points made available to the one-eyed stereo algorithm. Their correspondences were established by selecting an arbitrary origin and counting grid lines to obtain virtual image coordinates. When these pairs are processed by the algorithm in the appendix, a set of 3-D coordinates is obtained in either the viewer-centered coordinate space, or the virtual image coordinate space (which, if correct, is aligned with the original DTM). Figure 11(b) was produced from the resulting 3-D coordinates expressed in the virtual image space by using Smith's surface interpolation algorithm [16] to fit a surface to these points. This yields a dense set of 3-D coordinates that can then be displayed from any viewpoint. The viewpoint that was computed by one-eyed stereo was used to render the surface as shown in Figure 11(b). Its similarity to the original rendering (Fig. 11(a)) confirms the successful reconstruction of the scene.

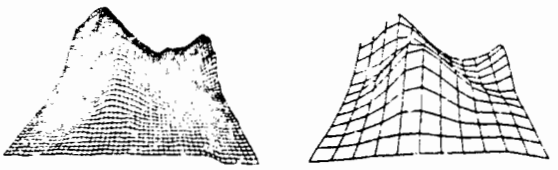


Figure 11: (a) View of part of a DTM (b) View of surface reconstructed from (a)

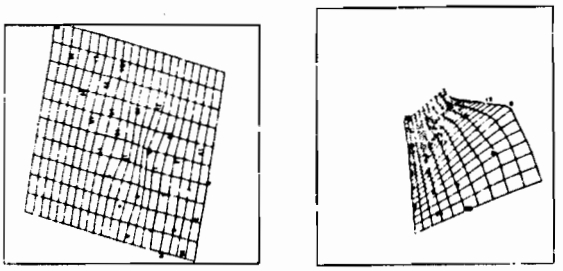


Figure 12: (a) Orthographic view of surface reconstructed from Figure 4 (b) Perspective view of same surface (from derived camera location)

The same procedure was followed when we worked with real photographs. The intersections of 31 street intersections were extracted manually from the photograph of San Francisco shown in Figure 4. Those that were occluded or indistinct were disregarded. Virtual image coordinates were obtained by counting city blocks from the lower-left intersection. The one-eyed stereo algorithm was then used to acquire 3-D coordinates of the corresponding image points in both viewer-centered and grid-centered coordinate systems. A continuous surface was fitted to both representations of these points. The location and orientation of the camera relative to the grid were also computed. Figure 12(a) shows the reconstructed surface as an orthographic view from the direction computed to be true vertical. The numbers superimposed are the computed locations of the original 31 points. Figure 12(b) shows the surface from the derived location of the viewpoint of the original photo. While several of the original points were badly mislocated, the general shape of the landform is apparent.

There are several reasons the algorithm can provide only a qualitative shape description. First, the problem itself can be somewhat sensitive to slight perturbations in the estimates of the piercing point or focal length. This appears to be inherent to the problem of recovering shape from a single image. How humans can perceive shape monocularly without apparent knowledge of the piercing point or semantic content of the scene remains unresolved. The second factor precluding precise, quantitative description of shape is the practical difficulty of acquiring large numbers of corresponding points. While the al-

gorithm can proceed with as few as eight points, the location of the object will be identified at those eight points only. If a more complete model is sought, additional points will be required to constrain the subsequent surface interpolation.

The task remains to evaluate the effectiveness of the iterative technique, described in Section 3.4, for recovering (1) shape from shading in the case of scenes possessing "unfriendly" reflectance functions, and (2) shape from nonprojective and distorted textures. Our experience with the process indicates that the key to surmounting these problems lies in the ability to establish valid correspondences with the virtual image. With these in hand, reconstruction of the surface can proceed as outlined in the foregoing discussion.

5 Conclusion

In this paper we have shown that, in principle, it is possible to employ the stereo paradigm in place of various approaches proposed for modeling 3-D scene geometry—including the case in which only one image is available. We have further shown that, for the case of a single image, the approach could be implemented by

(1) Setting up correspondences between portions of the image and some variants of a uniform grid, and;

(2) Treating each image region and its grid counterpart as a stereo pair, and employing a stereo technique to recover depth. (We present a new algorithm that makes it possible to accomplish this step.)

Devising automatic procedures to partition the image, select the appropriate form of the virtual image, and establish the correspondences are all difficult tasks that were not addressed in this paper. Nevertheless, we have unified a number of apparently distinct approaches, that individually, also have to contend with these same pervasive problems (*i.e.*, partitioning, model selection, and matching).

References

[1] Barnard, S. T., and Fischler, M. A., "Computational Stereo," Computing Surveys, Vol. 14, No. 4, December 1982.

[2] Brady, M., ed., *Artificial Intelligence* (Special Volume on Computer Vision), Volume 17, Nos. 1-3, August 1981.

[3] Cameron, R., *Above San Francisco*, Cameron and Company, San Francisco, 1976.

[4] Ganapathy, S., "Decomposition of Transformation Matrices for Robot Vision," International Conference On Robotics, (IEEE Computer Society), Atlanta, Georgia, March 13-15, 1984, pp. 130-139.

[5] Gennery, D. B., "Stereo Camera Calibration," Proceedings of the IU Workshop, November 1979, pp. 101-107.

[6] Horn, B. K. P., "Image Intensity Understanding," MIT Artificial Intelligence Memo 335, August 1975.

[7] Kender, J. R., "Shape from Texture," Ph.D. thesis, Carnegie-Mellon University, CMU-CS-81-102, November 1980.

[8] Lawton, D. T., "Constraint-Based Inference from Image Motion," Proc. AAAI-80, pp. 31-34.

[9] Longuet-Higgins, H. C., "A Computer Algorithm for Reconstructing a Scene from Two Projections," Nature, Vol. 293, September 1981, pp. 133-135.

[10] Nagel, H., and Neumann, B., "On 3-D Reconstruction from Two Perspective Views," Proc. IEEE 1981.

[11] Nitzan, D., Bolles, R.C., it et. al., "Machine Intelligence Research Applied to Industrial Automation," 12th Report SRI Project 2996, January 1983.

[12] Pentland, A. P., "Shading into Texture" Proceedings AAAI-84, August 1984, pp. 269-273.

[13] Prazdny, K., "Motion and Structure from Optical Flow," Proc. IJCAI-79, pp. 704-704.

[14] Roach, J. W., and Aggarwal, J. K., "Determining the Movement of Objects from a Sequence of Images," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 6, November 1980, pp. 554-562.

[15] Smith, G. B., "The Relationship between Image Irradiance and Surface Orientation," Proc. IEEE CVPR-83.

[16] Smith, G. B., "A Fast Surface Interpolation Technique," Proceedings: DARPA Image Understanding Workshop, October 1984, pp. 211-215.

[17] Stevens, K. A., "The Line of Curvature Constraint and the Interpretation of 3-D Shape from Parallel Surface Contours," AAAI-83, pp. 1057-1061.

[18] Stevens, K. A., "The Visual Interpretation of Surface Contours," Artificial Intelligence Journal Vol. 17, No. 1, August 1981, pp. 47-73.

[19] Strat, T. M., "Recovering the Camera Parameters from a Transformation Matrix," Proceedings: DARPA Image Understanding Workshop, October 1984, pp. 264-271.

[20] Tsai, R.Y. and Huang, T.S., "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-6, No. 1, Jan 1984, pp. 13-27.

[21] Ullman, S., *The Interpretation of Visual Motion*, The MIT Press, Cambridge, Mass., 1979.

[22] Witkin, A. P., "Recovering Surface Shape and Orientation from Texture," Artificial Intelligence Journal Vol. 17, No. 1, August 1981, pp. 17-45.

[23] Witkin, A., and Kass, M., "Analyzing Oriented Patterns," Proceedings IJCAI-85.

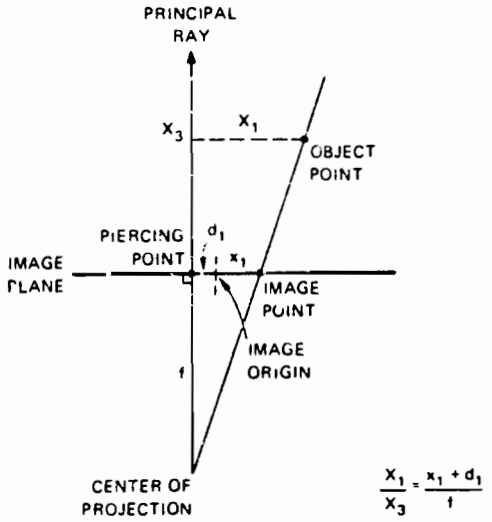


Figure 13: Definition of coordinate system

6 Appendix

The main body of this paper was devoted to showing how the problem of interpreting certain varieties of textured and shaded images can be transformed into equivalent problems in binocular stereo. Beginning with a perspective image, a second (virtual) image is hypothesized according to some presumed model of the original image. The model also specifies how to establish the correspondence between points in the two images. To compute the shape of the surfaces in the original scene, we need only compute the 3-D coordinates from the information in the two images, where the actual scene is a perspective projection and the virtual image has been constructed as an orthographic projection. This appendix shows how three-dimensional coordinates can be computed from point correspondences between a perspective and an orthographic projection when the relation between the imaging geometries is unknown.

We will use lowercase letters to denote image coordinates and uppercase letters for 3-D object coordinates. Unprimed coordinates will refer to the geometry of the perspective image, and primed coordinates to the orthographic image. Let x_1 and x_2 be the image coordinates of a point in the perspective image relative to an arbitrarily selected origin. Let $-d_1$ and $-d_2$ be the [unknown] image coordinates of the principal point and let f [> 0] be the focal length. The object coordinates associated with an image point are (X_1, X_2, X_3), where the origin coincides with the center of projection and the X_3 axis is perpendicular to the image plane. The X_3 coordinates of any object point will necessarily be positive.

The imaging geometry is as depicted in Figure 13 and yields the following standard perspective equations:

$$x_1 + d_1 = f\frac{X_1}{X_3}; \qquad x_2 + d_2 = f\frac{X_2}{X_3} \tag{3}$$

For the orthographic image, x'_1 and x'_2 are the image coordinates (relative to an arbitrary origin) and (X'_1, X'_2, X'_3) is the world coordinate system defined such that

$$x'_1 = X'_1; \qquad x'_2 = X'_2 \tag{4}$$

We use the unknown scale factor between orthographic image coordinates and the scene as our unit of measurement.

The two world coordinate systems can be related as follows:

$$X' = R(X - T) \tag{5}$$

where X is the column vector $[X_1,\ X_2,\ X_3]^T$,
X' is the column vector $[X'_1,\ X'_2,\ X'_3]^T$,
R is a 3x3 rotation matrix, and
T is a translation vector from the center of perspective projection to the origin of the world coordinate system associated with the orthographic projection. For either component (i=1 or 2), we can write

$$X'_i = R_i \cdot (X - T) \tag{6}$$

where R_i is the i-th row of R. By substituting Equations 3 and 4 into the above, we obtain

$$X_3 = \frac{f(x'_1 + R_1 \cdot T)}{R_1 \cdot [(x_1 + d_1),\ (x_2 + d_2),\ f]} \tag{7}$$

Eliminating X_3 from the two equations in Equation 7 yields

$$\begin{aligned} 0 = \ & x'_1 x_1 R_{21} + x'_1 x_2 R_{22} + x'_1 R_2 \cdot D \\ & - x'_2 x_1 R_{11} - x'_2 x_2 R_{12} - x'_2 R_1 \cdot D \\ & + x_1(R_{21} R_1 \cdot T - R_{11} R_2 \cdot T) + x_2(R_{22} R_1 \cdot T - R_{12} R_2 \cdot T) \\ & + R_1 \cdot T R_2 \cdot D - R_2 \cdot T R_1 \cdot D \end{aligned} \tag{8}$$

where D is the vector $[d_1,\ d_2,\ f]$.

The above equation relates image coordinates for corresponding points in both images. The following unknowns can be found by using eight corresponding pairs and solving the system of eight linear equations:

$$\begin{aligned} B_1 &= \tfrac{R_{21}}{R_{11}} \\ B_2 &= \tfrac{R_{22}}{R_{11}} \\ B_3 &= \tfrac{R_2 \cdot D}{R_{11}} \\ B_4 &= \tfrac{R_{12}}{R_{11}} \\ B_5 &= \tfrac{R_1 \cdot D}{R_{11}} \\ B_6 &= \tfrac{R_{21}}{R_{11}} R_1 \cdot T - R_2 \cdot T \\ B_7 &= \tfrac{R_{22}}{R_{11}} R_1 \cdot T - \tfrac{R_{12}}{R_{11}} R_2 \cdot T \\ B_8 &= \tfrac{1}{R_{11}}(R_1 \times T) \times (R_2 \cdot D) - \tfrac{1}{R_{11}}(R_2 \cdot T)(R_1 \cdot D) \end{aligned} \tag{9}$$

$$\begin{bmatrix} x'_1 x_1 & x'_1 x_2 & x'_1 & -x'_2 x_2 & -x'_2 & x_1 & x_2 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \end{bmatrix} = \begin{bmatrix} x'_2 x_1 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \tag{10}$$

When more than eight points are available, a least-squares method can be employed to solve the system of equations. Once we have the B_i's in hand, we can solve for the components of the rotation matrix R. First, R_{11} can be determined by making use of the fact that the rows of a rotation matrix are orthogonal. Thus, from $R_1 \cdot R_2 = 0$ and the expressions for B_1, B_2 and B_4 in Equation 9, we get

$$R_{11}^4(B_4^2 B_1^2 + B_2^2 - 2B_1 B_2 B_4) - R_{11}^2(1 + B_1^2 + B_2^2 + B_4^2) + 1 = 0 \tag{11}$$

This yields two real values for F_{11}; fortunately we'll be able to identify the incorrect one later. For now, let us simply choose one at random and return to this point if it turns out to be wrong.

The rest of R can be derived from the B_i's in a similar fashion. R_{12}, R_{21} and R_{22} can be established immediately from R_{11} and Equation 9. R_{13} is determined from the fact that $\| R_1 \| = 1$. $R_1 \cdot R_2 = 0$ gives an expression for R_{23}. Finally, R_3 is computed from the fact that $R_1 \times R_2 = R_3$ for all rotation matrices. As a result, we have completely derived two alternative R matrices, depending on the choice of R_{11}. One of these matrices is correct, while the other can be eliminated later.

Now to solve for the translation vector T. First let us note that T cannot be found uniquely, because the origin of the primed world coordinate system has not been completely specified. The X'_1 and X'_2 coordinates of the origin were fixed by the choice of origin for the orthographic image coordinates, but the position of the origin along the X'_3 axis is still unconstrained. Since we are free to choose any origin for X', we will choose the one for which $T_3 = 0$.

Using the expression for B_6 in Equation 9, we find

$$B_6 = \frac{R_{21}}{R_{11}}(R_{11}T_1 + R_{12}T_2 + R_{13}T_3) - (R_{21}T_1 + R_{22}T_2 + R_{23}T_3) \tag{12}$$

Making use of the fact that $R_{33} = R_{11}R_{22} - R_{12}R_{21}$ and $T_3 = 0$, we get

$$T_2 = -B_6\frac{R_{11}}{R_{33}} \tag{13}$$

Similarly,

$$T_1 = B_7\frac{R_{11}}{R_{33}} \tag{14}$$

The origin of the primed coordinate system in unprimed coordinates is given by

$$T = [B_7\frac{R_{11}}{R_{33}},\ -B_6\frac{R_{11}}{R_{33}},\ 0]. \tag{15}$$

If the location of the principal point is known but the focal length (the scale factor of the perspective image) is not, f can easily be computed from Equation 9:

$$f = \frac{B_5R_{11} - R_{11}d_1 - R_{12}d_2}{R_{13}} \tag{16}$$

If the focal length is known, the principal point of the perspective image is found as follows. Use the third and fifth expressions of Equation 9 to write two equations in the two unknowns, d_1 and d_2. Their solution yields

$$\begin{aligned} d_1 &= f\frac{R_{11}}{R_{13}} + \frac{B_5R_{11}R_{23} - B_7R_{11}R_{13}}{R_{33}} \\ d_2 &= f\frac{R_{12}}{R_{13}} + \frac{B_7R_{11}^2 - B_5R_{11}R_{21}}{R_{33}} \end{aligned} \tag{17}$$

The perspective image coordinates of the principal point are $[-d_1,\ -d_2]$.

If neither the focal length nor principal point is known beforehand, then the problem we have proposed does not have a unique solution. Equation 17 specifies the constraints between focal length and piercing point. For any choice of focal length, there exists a unique principal point. The center of perspective projection is constrained to lie on a line parallel to the lines of sight of the orthographic projection. The reconstructed surface will be distorted as one varies the center of projection along this line. It is worth noting, however, that our computations of the rotation matrix R and the translation vector T did not require knowledge of either the focal length or the principal point.

We are now in a position to compute the world coordinates of all points for which we have correspondences. There may, of course, be many more than the minimum of eight points used so far. Equation 6 gives

$$x'_1 = R_1 \cdot [\frac{X_3}{f}(x_1 + d_1),\ \frac{X_3}{f}(x_2 + d_2),\ X_3] - R_1 \cdot T \tag{18}$$

which can be solved for

$$X_3 = \frac{f(x'_1 + R_1 \cdot T)}{R_{11}x_1 + R_{12}x_2 + R_1 \cdot D} \tag{19}$$

Now we must check the signs of the X_3's. If they are negative, the world points are located behind the center of projection. The correct solution, corresponding to all positive values of X_3, can be found by choosing the alternative value of R_{11}, derived earlier and repeating the computations from that point. After obtaining the set of positive X_3's, we can continue.

Equation 3 gives the other unprimed world coordinates:

$$X_1 = \frac{X_3}{f}(x_1 + d_1); \qquad X_2 = \frac{X_3}{f}(x_2 + d_2) \tag{20}$$

If desired, the primed coordinates can be found by applying Equation 5.

The above derivation makes the implicit assumption that the X'_1 and X'_2 axes are scaled equally. It is conceivable that the virtual image coordinates could be unequally scaled, as is the case when they are derived from a rectangular grid (*e.g.*, Figure 2). If we have prior knowledge of the ratio of the sides of each rectangular grid element, then the virtual image coordinates should be normalized before applying this algorithm (*i. e.*, by dividing X'_2 by this ratio). Without knowledge of the ratio, the problem is underspecified and a unidimensional class of solutions exists. Knowledge of the piercing point, if available, can be used to constrain the problem further and to solve for the unique solution. To do this, we define the following virtual coordinate systems in place of Equation (4):

$$x'_1 = X'_1; \qquad x'_2 = \frac{1}{k}X'_2 \tag{21}$$

where k is the ratio of the sides of the rectangular grid elements.

The solution proceeds as before, yielding

$$\begin{aligned} 0 = \ & x'_1x_1R_{21} + x'_1x_2R_{22} + x'_1R_2 \cdot D \\ & -kx'_2x_1R_{11} - kx'_2x_2R_{12} - kx'_2R_1 \cdot D \\ & + r_1(R_{21}R_1 \cdot T - R_{11}R_2 \cdot T) + r_2(R_{22}R_1 \cdot T - R_{12}R_2 \cdot T) \\ & + R_1 \cdot TR_2 \cdot D - R_2 \cdot TR_1 \cdot D \end{aligned} \tag{22}$$

The above equation is recast as the eight linear equations:

$$\begin{bmatrix} -x'_1x_2 & -x'_1 & x'_2x_1 & x'_2x_2 & x'_2 & x_1 & x_2 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \end{bmatrix} = \begin{bmatrix} x'_1x_1 \\ \cdot \\ \cdot \end{bmatrix} \tag{23}$$

where

$$
\begin{aligned}
C_1 &= \frac{R_{22}}{R_{21}} \\
C_2 &= \frac{R_{23}}{R_{21}} D \\
C_3 &= \frac{R_{11}}{R_{21}} k \\
C_4 &= \frac{R_{12}}{R_{21}} k \\
C_5 &= \frac{R_{13}}{R_{21}} k D \\
C_6 &= \frac{R_{11}}{R_{21}} R_2 \cdot T - P_1 \cdot T \\
C_7 &= \frac{R_{12}}{R_{21}} R_2 \cdot T - \frac{R_{22}}{R_{21}} R_1 \cdot T \\
C_8 &= \frac{f}{R_{21}} (R_2 \cdot T)(R_1 \cdot D) - \frac{1}{R_{21}} (R_1 \cdot T)(R_2 \cdot D)
\end{aligned}
\tag{24}
$$

The following equalities can then be derived from Equation (24):

$$
\begin{aligned}
R_{13} &= \frac{R_{21}}{f k} (C_5 - C_3 d_1 - C_4 d_2) \\
R_{23} &= \frac{R_{21}}{f} (C_2 - d_1 - C_1 d_2)
\end{aligned}
\tag{25}
$$

$$
f = \sqrt{\frac{-(C_5 - C_3 d_1 - C_4 d_2)(C_2 - d_1 - C_1 d_2)}{C_3 + C_1 C_4}} \tag{26}
$$

$$
R_{21} = \pm \frac{f}{\sqrt{f^2 + C_1^2 f^2 + (C_2 - d_1 - C_1 d_2)^2}} \tag{27}
$$

$$
k = \frac{R_{23}}{f} \sqrt{f^2 C_3^2 + f^2 C_4^2 + (C_5 - C_3 d_1 - C_4 d_2)^2} \tag{28}
$$

The rest of R can now be computed easily from Equation (24) and $R_1 \times R_2 = R_3$. The translation vector T is given by Equation (15) because $C_6 = B_6$ and $C_7 = B_7$. With R and T now fully recovered, it is a simple matter to derive the object coordinates from Eqs. (3), (21), and (5). Let us recall that we have two candidate matrices R hinging on the choice for R_{21}; as before, the correct one must be selected by examining the signs of the X_3 coordinates.

To summarize, we have described an algorithm to compute the relative orientation and position between two imaging systems—perspective and orthographic—from the locations of eight (or more) corresponding image points. Either the principal point or the focal length and rectangular aspect ratio are computed along the way. With this information in hand, the world coordinates of all points in the imaged scene can be computed.

GC970145-40

STEREO CORRESPONDENCE: FEATURES AND CONSTRAINTS

Hong Sch Lim & Thomas O. Binford

Artificial Intelligence Laboratory, Computer Science Department
Stanford University, Stanford, CA 94305, USA.

ABSTRACT

This is a preliminary report on a high level stereo system for recovering depth of a three dimensional scene from a stereo pair of images. Junctions, edgels, extended curves and regions are features used for matching. Geometrical properties of features and relations among features are used as constraints to guide both local matching and global matching. The result is a sparse disparity map of the analyzed scene.

Junctions are determined by extending curves and using intensity values from image. They are classified and used for matching. We illustrate how this system can solve two classical correspondence problems. We show results of correspondence for line drawings and preliminary results from real images. These results indicate that this technique renders not only an accurate local match but also a globally consistent match. The computation complexity is reduced by two orders of magnitude when compared to some existing methods.

INTRODUCTION

A high level stereo system tries to match structures derived from pair of stereo images. Structures are extracted from image by effective segmentation and aggregation. We try to infer surfaces and three-dimensional structures by interpreting structures from images [Binford 1981, Malik 1984]. We believe using high level structures for matching will give a more reliable and globally consistent correspondence.

Stereo matching has the advantage of being a passive method of ranging. It has been applied to various domains, e.g. obstacle avoidance in navigation [Moravec 1980], aerial cartography [Panton 1978], automatic surveillance [Henderson 1979] and modeling objects for model-based vision [Takamura 1984].

In stereo matching we want to identify corresponding views of the same element in the physical scene. Changes in view point change the images of these corresponding elements. Features chosen for matching should therefore be invariant or vary slowly with change of view point. A difficulty is to achieve globally consistent matching. Local features or areas in one image may match with more than one feature or area in the other image. These ambiguities in local matches can be resolved by incorporating global strategies. The key problems in stereo matching are therefore choosing the features to be matched, determining constraint relations, and designing the strategies for matching these features both locally and globally.

First we give a brief review of previous work in stereo matching, then we discuss our approach to the problem. Our technique is based on high level constraints which require high quality input. The Nalwa edge segmenter [Nalwa 85] provides extended curves and junctions. As part of this stereo system, we have improved junction determination significantly. We test our technique on some curve drawings and then present some preliminary results of applying this technique to real images. An analysis of the complexity of computation is also given.

REVIEW

Two principal techniques have been used in stereo matching, area-based matching [Hannah 1974, Panton 1978] and feature-based matching [Baker 1981, Medioni 1985].

Area-based matching tries to match an area of pixels in one image to another image. Ideally one would like to locate a correspondence pixel for each pixel in both images. But the information in one pixel is not enough to resolve the ambiguity in matching. As a result, a small window is chosen as the matching unit. A window in one image is matched with a range of windows in the other image using cross-correlation or similar measure of the similarity between two windows.

Area-based matching has been applied with partial success in terrain mapping. However, it requires the presence of detectable texture for matching; it tends to break down when the scenes have textureless areas, repetitive texture, or surface discontinuities. At surface discontinuities there are occlusions and no possible correspondence between areas crossing the occlusion. The accuracy for correspondence in area-based matching depends on the window size and is generally an order of magnitude less than that of feature-based matching. Computation time can be reduced by matching only areas that are of particular interest, e.g. with large variance.

Features such as junctions or edges are extracted from the intensity image for matching. The underlying principle is that a discontinuity in the intensity represents a discontinuity on the physical surfaces in the scene. The discontinuity can be due to surface depth, orientation, reflectance, or illumination. All these curve discontinuities occur at points on the physical surfaces. By matching these features we match physical curves on object surfaces, except at limbs.

The location of curve features in an image can be estimated to sub-pixel accuracy [MacVicar-Whelan 1981]. The accuracy of the recovered three dimensional depth is higher than that of area-based matching. Note that the number of features is in general less than the number of pixels. As a result, the computation time is less. Since not every point in an image corresponds to a feature, feature matching leads only to a sparse depth map. To produce a dense depth map, it must be accompanied by model-based interpretation, surface interpolation, or by area matching.

The search space for matching features can be greatly reduced by knowing the geometric relationships between the cameras used in taking the images. Family of planes passing through the object and the two camera foci are called epipolar planes. Epipolar lines are the intersection of the epipolar planes with the image planes. Any image point lying on a particular epipolar line will find its corresponding points on the corresponding epipolar line in the other image.

In particular, if we restrict the camera geometry to be such that the principal horizon curves of both images are collinear and the principal vertical curves of both images are parallel, that is the two cameras are related by a horizontal displacement, then the epipolar lines are just the horizontal curves in the image. The search for correspondence points will be limited to the same horizontal curve. Locations of features can be transformed into the canonical stereo system [Gennery 1979].

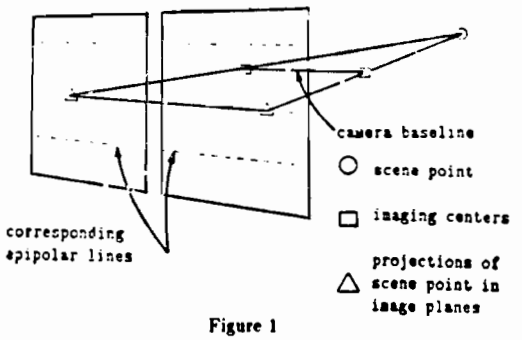


Figure 1

Perkins [Perkins 1970] pointed out the difficulties involved in trying to resolve the matching problem without resource to higher level information. A hierarchical scheme for matching stereo images of polyhedra was implemented by Ganapathy [Ganapathy 1976]. He also studied various rules for stereo matching. A cooperative computation algorithm was proposed by Marr, Poggio and Grimson [Marr 1976, 1977, Grimson 1979] which matches random dot stereograms. It uses uniqueness constraint to assign at most one disparity value to each point in the image and continuity constraint to require the disparity to vary smoothly, except at depth discontinuities. Arnold and Binford [Arnold 1978] used edge orientation, intensity and edge continuity to determine a set of globally optimal matches. They [Arnold 80] also introduced quasi-invariants for correspondence of edges and surface normals. Baker and Binford [Baker 81] match edges on epipolar lines using those quasi-invariants and uses dynamic programming to preserve the order of edges. A connectivity constraint was used to removed globally inconsistent edge correspondences. Ohta and Kanade[Ohta 1983] extended Baker's method from intra-scanline search to inter-scanline search which take into account of the mutual dependency between epipolar lines in an image. Another system which uses segments, groups of collinear connected edge points, as matching primitive, was implemented by Medioni and Nevatia [Medioni 1985]. Its correspondence is based on a minimum differential disparity criterion.

OUR METHOD

We have chosen junctions, edgels, curves, and regions to be our matching features. Junction are end points of curves and the intersections of curves. The degree of a junction is the number of curves forming the junction. Junctions of degree three or less can be classified as I (junction of degree one), L, Y, A, and T. Curves are a connected group of edgels; each curve has two junctions. The very fact that these features are chosen is because they are quasi-invariant to view point. That is with a change of view point, these features

remain unchanged in the images except under accidental alignment. Also all these features correspond to the physical points in the scene, with the exception of a T junction which is normally formed by occlusion. Another exception is the limb of any curved object which forms a curve in the image but the location of this curve on the object changes with view points. Therefore, we would like to distinguish limbs from real edges and T junctions from other types of junctions.

Our method matches not only junctions, edgels, curves, and regions but also relations between them. Note that the junctions of a matched pair of curves must match and the curves between two pairs of matched junctions must also match.

Our technique determines all possible matches of junctions lying on one epipolar line (or within an equivalence class of epipolar lines depending on measurement error). We first assert that the type and degree of the chosen junctions are matched. Though the type and degree of the junctions are not view point invariant, this is a good way to prune off unwanted match. As for those matches that are wrongly pruned away, we will come back later and match them separately. We want to point out that in fact both the type and degree of the junctions can vary widely under wide angle stereo.

In order to assert a particular match for a junction, all the junctions which are connected to this junction by curves must be matched unless occluded. The contrast and intensity across curves joining these junctions must also be consistent. By so doing, we implicitly enforce a constraint for matching globally. Uniqueness in matching is enforced in our method. Each point in an image may only be matched to at most one point in the other image. In effect, we do not allow transparent objects in our images. This system now deals with occlusion indicated by T junctions and limbs but not those occlusions where surfaces disappear.

After the initial matching, we come back to those curves that are still unmatched. For those curves that have only one end point matched, we start matching from the matched junction and trace the curve trying to match every edgel of the curve to the corresponding curve in the other image. Again we use the grey scale intensity and contrast across the edge, direction along the edge and the epipolar constraint to confirm the match.

In essence, when we match junctions with the global constraint, we actually nail down some points in the image. Using these matched points as reference, we try to match all points between.

ON LINE DRAWINGS

We will demonstrate the effectiveness of the above techniques using some curve drawings. Part of the technique can be implemented as an integer programming problem, or using other constraint systems.

For every junction i on the left image A we try to find a junction j in the right image B along the same epipolar line. For each pair we created a variable ABij. Later we will assign ABij to be 1 if it is a match, 0 if it is not.

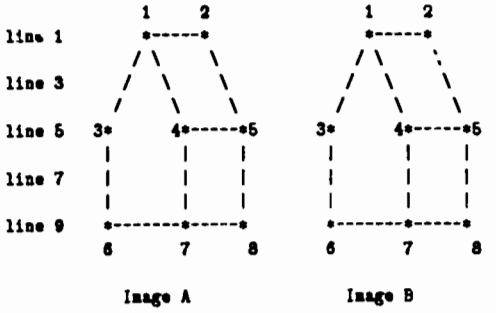


Figure 2

For example, from the first epipolar line we create variables AB11, AB12, AB21 and AB22 for all the possible matches of the junctions 1 and 2 in image A to junctions 1 and 2 in image B. For each junction we want to have all junctions connecting to it to be matched also. That is, to match junction 1 in image A to junction 1 in image B, we require that junctions 3, 4 and 2 in the left image match those in the right image. It is represented by:

$$3ab11 - ab22 - ab33 - ab44 <= 0.$$

This says that the order of junction 1 is three and we need all the connected junctions, namely 2, 3 and 4, to match in order for junction 1 to be matched. In the meantime, we are also implicitly matching the order of the junctions. Finally the condition that the matching is unique is added. For example:

$$ab11 + ab12 = 1$$

which implies that junction 1 in the left image can only be matched to junction 1 or 2 in the right image but not both. In this example we did not add in the constraints that the contrasts across the curves have to be matched. We want to maximize the evaluation function which is the total number of matched junctions. The results indicate that all the junctions are matched correctly (see appendix for the details of the program). Consequently, the curves between junctions can be matched in no time.

The effectiveness of this technique can also be demonstrated by the fact that it can solve two of the classical problems encountered in stereo matching.

Consider the case of matching images of a checker board or any other image with a repetitive pattern. All the junctions in the inner part of the image are of the same type and degree. Any matching strategy based on local constraints is bound to come up with ambiguous matches.

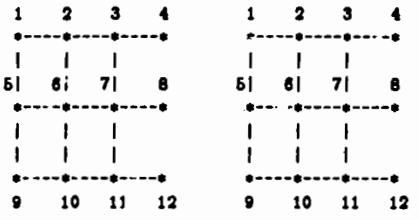


Image A Image B

Figure 3

Junctions A6, A7 in image A and junctions B6, B7 in image B have the same type and degree. The grey scale and contrast nearby can also be the same. They cannot be distinctively matched with only local information. With global constraints, we note that the order of A5 connecting to A6 is different from the order of B6 connecting to B7. Consequently, A6 cannot be matched to B7. The same reasoning holds for A7 and B6. Therefore the ambiguity no longer exists.

The above problem can sometimes be solved by matching the order of features along an epipolar line. Though it is a good strategy and it holds for most of the time, order reversal does occur. The following example illustrates that our technique can overcome the problem of order reversal.

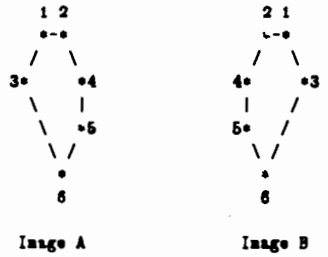


Image A Image B

Figure 4

Ai will match all Bi (i = 1,...,6) using this matching technique.

Note that junction A3 is to the left of junction A4 in image A but B3 is to the right of junction B4. The reason that A4 is matched to B4 is that A2 and A5 connecting to A4 match with B2 and B5 of B4. And we reject the match of A3 and B4 by the fact that A6 connecting to A3 does not match with B5 connecting to B4.

ON REAL IMAGES

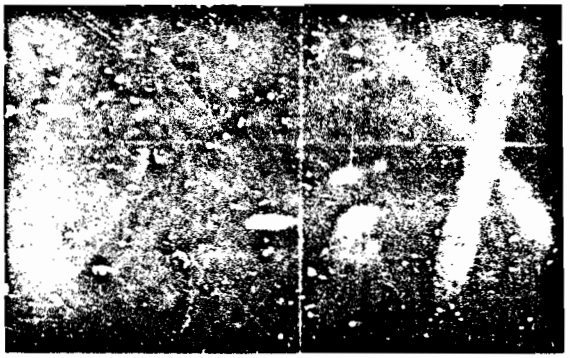
The curve segments used are obtained by applying the Nalwa Operator [Nalwa 1984] to a pair of stereo images. The operator produces a list of edgels by fitting a directional tanh-surface to windows in the images. These edgels are then linked and fitted with conic sections and straight lines [Nalwa 1985]. The linking and curve fitting algorithms also locate L junctions which we have utilized.

We have modified our algorithm to deal with data from real images. In real images the features detected include additional spurious edges and missing edges. Some curves appear broken. Also the type and degree of a junction may differ from a noise free line drawing.

Therefore we want to bridge broken curves and reconstruct junctions to the greatest extent before we do the matching. We extend curves along their tangent directions. The intensity on both sides of the curve, obtained from image, is used to guide extending the edge. Since we know the approximate position and direction of the extended edge, we plan to use a small operator to pinpoint the location of the extended edge after a larger operator to locate unknown edges. This will give a more accurate localization of edges that could not be located before. We find some new junctions when the extended edge intersects another curve. The type and order of a junction will change when the extended edge runs into that junction.

All junctions are classified by their type and order before the matching phase. However, we feel that a good corner finder is essential to find all the junctions and locate them more accurately. Existing corner finders essentially search for places that have a high curvature in the intensity surface. Many of which are really not junctions in the image [Dreschler 1982, Kitchen 1982].

We then use the above algorithm to match all junctions that can be matched. We relax the constraint that all connected junctions must be on the same epipolar lines. They may have different degree or type. In noisy images some junctions may be distorted because of missing curves. Also, we require only that majority of the connected junctions be matched. A broken curve has different end points compared to its corresponding curve. Broken curves are matched point by point by tracing along the curves. Relaxing the requirement



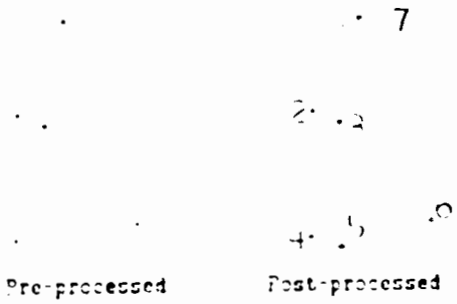
Left Image

Right Image

Figure 5

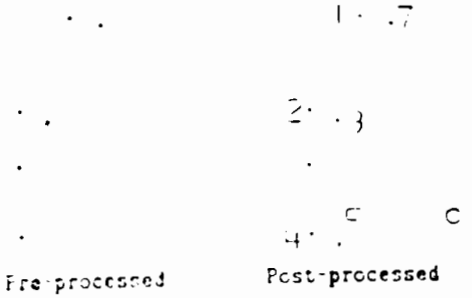
Figure 6

Figure 7



Pre-processed Post-processed

Figure 8



Pre-processed Post-processed

Figure 9

also means that the global constraint is not as strong as before. However, our results shows that relaxing the constraints does not cause significant deterioration.

In figure 5 we show a pair of grey scale images of some blocks, cylinder and spheres. Figure 6 is the result from edge detection and edge linking. Note that most of the junctions are not formed properly. The main reason is that the edge detection algorithm is designed to look for a single step edge in a window. It breaks down when there are more than one edge in the window.

Figure 7 shows the results obtained by extending the curves. Note that the missing gap on the cylinder is bridged. Some new junctions are found and some old ones are modified as their degree and type change. An enlargement of one of the blocks is shown in figure 8 and 9. On the left are the pre-processed drawings and on the right are the post processed images.

Our algorithm matches all junctions except junction 6 in the left image and junction 7 in the right (post-processed images of figure 8 and 9). Note that in matching curves between junctions 2 and 4, we bypass the T junction between since it corresponds to an occlusion junction. We are unable to match junction 6 because a corresponding L junction is not found in the right image. Instead, a high curvature conic curve is fitted to that junction. Failure to match junction 7 is due partly to the above reason and partly to the fact that the extended curves did not form a new junction. Matching of the curves with both end points matched are simple. We match the other unmatched curves starting from the matched junctions and trace along both curves as described above.

COMPUTATION TIME

We believe that by matching the junctions first and then edgels along curves between junctions leads to efficient matching. The number of junctions in an image is much less than the number of edgels in the same image and the information at a junction is more than that in an edgel.

We analyze the complexity of matching by edgels first and then compare it with the methods of matching junctions.

Assume there are

E	- number of edgels in an image
N * N	- number of pixels in an image
L	- average number of edgels in a curve

Therefore, on the average,

E/N	- number of edgels per scan line
$(E/N)^2$	- pairs of edgels per scan lines for matching
E^2/N	- pairwise combinations of edgels which satisfy epipolar constraint for whole image
E/L	- number of curves in an image
V = 2E/L	- number of junctions (worse case where no two lines have the same end points)
V/N	- number of junctions per scan line
$(V/N)^2$	- pairs of junctions per scan lines for matching
V^2/N	- pairwise combinations of junctions which satisfy epipolar constraint for whole image

The total combination for matching junctions is

$V^2/N = (4/L^2)*(E^2/N)$,

which is a factor of $4/L^2$ less than that of matching edgels.

When the length of the curve segment is on average 30 pixels long then there is a 225 times reduction in computation time. However, we must balance the cost of the stereo correspondence with the cost of segmentation and aggregation required to get the required additional structures before matching.

CONCLUSION

We believe our technique offers an efficient and reliable way for stereo matching. It matches not only the chosen features themselves using local constraints but also the relationships between these features by global constraints. However, we do have to modify the technique in the presence of noise. This is a preliminary report. We are incorporating many more constraints at all structural levels. The system will be applied to more images derived from the real world and more results will follow.

ACKNOWLEDGEMENT

This work was supported by the Defense Advanced Research Projects Agency under contract Number N 0039-84-C-0211. The authors would like to extend their gratitude to Vishvjit S. Nalwa and Eric Paulson for providing the necessary data for this work.

REFERENCE

[Arnold 1978] D.Arnold, "Local context in matching edges for stereo vision", Proceeding: Image understanding workshop, May, 1978.

[Baker 1981] H.H. Baker, "Depth from edge and intensity based stereo", Ph.D. thesis, University of Illinois, September, 1981.

[Binford 1981] T.O.Binford, "Inferring surfaces from images", Artificial Intelligence, Volume 17, 1981.

[Dreschler 1982] L. Dreschler, H.H.Nagel, "Volumetric model and 3D trajectory of a moving car derived from monocular TV frame sequences of a street scene", Computer Graphics and image processing, Volume 20, 1982.

[Hannah 1974] M.J. Hannah, "Computer matching of areas in stereo imagery", Ph.D. thesis, Stanford University, 1974.

[Ganapathy 1976] S. Ganapathy, "Reconstruction of scenes containing polyhedra from stereo pair of views", Ph.D. thesis, Stanford University, 1976.

[Gennery 1979] D. Gennery, "Stereo camera calibration", Proceeding: Image understanding workshop, November, 1979.

[Grimson 1979] W.E.L.Grimson, D.Marr, "A computer implementation of a theory of human stereo vision", Proceeding: Image understanding workshop, April, 1979.

[Henderson 1979] R.L. Henderson, W.J. Miller, C.B Grosch, "Automatic stereo reconstruction of man-made targets", Society of photo-optical instrumentation engineer, Volume 186, Number 6, 1979.

[Kitchen 1982] L.Kitchen, A.Rosenfeld, "Gray-level corner detection", Pattern recognition letters, Volume 1, 1982.

[MacVicar-Whelan 1981] P MacVicar-Whelan, T Binford, "Curve finding with subpixels precision", Proceeding: Image understanding workshop, April, 1981.

[Malik 1984] J Malik, T O Binford, "A theory of line drawing interpretation", Proceedings: Image understanding workshop, October, 1984.

[Marr 1976] D Marr, T Poggio, "Cooperative computation of stereo disparity", Science, Volume 194, 1976.

[Marr 1977] D Marr, T Poggio, "A theory of human stereo vision", MIT AI Memo, number 451, November, 1977.

[Medioni 1985] G.M. Medioni, R. Nevatia "Segment-based stereo matching", Computer vision, graphics, and image processing, Volume 31, 1985.

[Moravec 1980] H.P. Morravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover", Ph.D. thesis, Stanford University, May, 1980.

[Nalwa 1984] V. S. Nalwa, "On detecting edges", Proceedings: Image understanding workshop, October, 1984.

[Nalwa 1985] V. S. Nalwa, E. Pauchon, "Algorithms for edge-aggregation and edge-description", Proceeding: Image understanding workshop, December, 1985.

[Ohta 1983] Y.Ohta, T.Kanade, "Stereo by intra and inter-scanline search using dynamic programming", Technical report CMU-CS-83-162, October, 1983.

[Panton 1978] D.J. Panton, "A Flexible approach to digital stereo mapping", Photogrammetric Engineering and remote sensing, Volume 44, Number 12, December 1978.

[Perkins 1970] D.N.Perkins, "Computer stereo vision: A combinatorial theory with implementation", Ph.D. thesis, Department of Mathematics, M.I.T., June 1970.

[Takamura 1984] J.Takamura, T.O. Binford, "Stereo modeling system: A geometric modeling system for modeling object instance and class", Proceedings: Image understanding workshop, October, 1984.

APPENDIX

The objective function says that we want to maximize the number of matched junctions.

```
max ab11 + ab12 + ab21 + ab22 + ab33 + ab34 + ab35 + ab43
  + ab44 + ab45 + ab54 + ab56 + ab65 + ab66 + ab67 + ab68
  + ab76 + ab77 + ab78 + ab86 + ab87 + ab88
st
```

(The following inequalities require that the junctions can only be matched if all the junctions connecting to it are also matched)

```
   3ab11 - ab22 - ab33 - ab44 <= 0
   2ab22 - ab11 - ab55 <= 0
   2ab33 - ab11 - ab66 <= 0
   3ab44 - ab11 - ab55 - ab77 <= 0
   3ab45 - ab12 - ab54 - ab78 <= 0
   3ab54 - ab21 - ab45 - ab87 <= 0
   3ab55 - ab22 - ab44 - ab88 <= 0
   2ab66 - ab33 - ab77 <= 0
```

```
    2ab68 - ab56 - ab77 <= 0
    3ab77 - ab44 - ab56 - ab88 <= 0
    2ab88 - ab53 - ab77 <= 0
    2ab88 - ab55 - ab77 <= 0

(The following equations establish the uniqueness
 constraints)

    ab11 + ab12 = 1
    ab21 + ab22 = 1
    ab33 + ab34 + ab35 = 1
    ab43 + ab44 + ab45 = 1
    ab53 + ab54 + ab55 = 1
    ab66 + ab67 + ab68 = 1
    ab76 + ab77 + ab78 = 1
    ab78 + ab87 + ab88 = 1
    ab11 + ab21 = 1
    ab12 + ab22 = 1
    ab33 + ab43 + ab53 = 1
    ab34 + ab44 + ab54 = 1
    ab35 + ab45 + ab55 = 1
    ab66 + ab76 + ab86 = 1
    ab67 + ab77 + ab87 = 1
    ab68 + ab78 + ab88 = 1

(The following equations make sure that there is either a
 match or no match but nothing between)

    abij = 1 or 0 (i = 1, ...,8; j = 1,...,8)

(Results from LINDO, a linear programming package which
 has the option of solving integer programming problem)

    abij = 1 for i = j

    abij = 0 for i <> j
```

GC970145-41

Direct Passive Navigation: Analytical Solution for Planes

S. Negahdaripour and B.K.P. Horn

The Artificial Intelligence Laboratory, Massachusetts Institute of Technology

In this paper, we derive a closed form solution for recovering the motion of an observer relative to a planar surface directly from image brightness derivatives. We do not compute the optical flow as an intermediate step, only the spatial and temporal intensity gradients at a minimum of 8 points. We solve a linear matrix equation for the elements of a 3x3 matrix. The eigenvalue decomposition of its symmetric part is then used to compute the motion parameters and the plane orientation.

Introduction

The problem of determining rigid body motion and surface structure from image data has been the topic of many research papers in the area of machine vision [1-22]. Many approaches based on, tracking feature points [5,11,19,20] or contours [9], using motion flow field [1,3,4,10,12,16,17,21,22] texture [2], or image intensity gradients [14,15] have been proposed in the literature.

In the feature point matching schemes, information about a finite number of well-seperated points is used to recover the motion (general 8-point 2-frame algorithms of Longuet-Higgins [11], Tsai and Huang [20], Buxton et al. [5], and the algorithm of Tsai, Huang and Zhu [19] for planar surfaces). These methods require identifying and matching feature points in a sequence of images. The minimum number of points required depends on the number of image frames. With 2 frames, in most cases, a minimum of 5 points results in a unique solution from a set of nonlinear equations. However, using 8 points, as in algorithms cited above, one only solves linear equations. Here, it is assumed that the more difficult problem of establishing point correspondence has already been solved. In general, this involves determining corners along contours using iterative searches. For images of smooth objects, it is difficult to find good features or corners.

For the general case of smooth surfaces, Longuet-Higgins and Prazdny [11] suggested a method that uses the optical flow and its first and second derivatives at a single point. Later, Waxman and Ullman [21] developed this into an algorithm for recovering the structure and motion parameters from a set of nonlinear equations. Subbarao and Waxman [17] recently found a closed form solution to the original formulation in [21] for planar surfaces. These methods, while mathematically elegant, are very sensitive to errors in the optical flow data since second order derivatives of noisy data are used.

At the expense of more computation, more robust algorithms have been suggested using the optical flow at every image point [1,3,4]. Longuet-Higgins [12] has presented a closed form solution for planar surfaces, very similar to ours, using the coefficients of the second order optical flow equations. However, it is assumed that both components of the flow field have already been computed for a minimum of 5 image points.

By representing a planar surface in the form of a closed contour, Kanatani [9] has shown that the surface and motion parameters can be computed by measuring "diameters" of the contour using line and surface integrals. Here, no point correspondence is required. Assuming that the planar surface has a uniform texture density, Aloimonos and Chou [2] have presented a procedure for computing the motion and surface orientation from texture.

In much of the research work in recovering surface structure and motion from the optical flow field, it is assumed that a reasonable estimate of the full optical flow field is available. In general, the computation of the local flow field exploits a constraint equation between the local intensity changes and the two components of the optical flow. However, this only gives the component of the flow in the direction of the intensity gradient. To compute the full flow field, one needs additional constraints such as the heuristic assumption that the flow field is locally smooth [7,8]. This, in many cases, leads to optical flow fields that are not consistent with the true motion field.

In an earlier paper, we presented an iterative scheme for recovering the motion of an observer relative to a planar surface directly from the image brightness derivatives, and the need to compute the local flow field [14,15]. Further, using a compact vector notation, we showed that, at most, two interpretations are possible for planar surfaces and derived the relationship between them. Here, we present a closed form solution to the same problem. We first solve a linear matrix equation for the elements of a 3x3 matrix using intensity derivatives at a minimum of 8 non-colinear points. The special struc-

ture of this matrix allows us to compute the motion and structure parameters very easily.

2. Preliminaries

We first recall some details about perspective projection, the motion field, the brightness change constraint equation, rigid body motion and planar surfaces. This we do using vector notation in order to keep the resulting equations as compact as possible.

2.1. Perspective Projection

Let the center of projection be at the origin of a Cartesian coordinate system. Without loss of generality we assume that the effective focal length is unity. The image is formed on the plane $z = 1$, parallel to the xy-plane, that is, the optical axis lies along the z-axis. Let $\mathbf{R}$ be a point in the scene. Its projection in the image is $\mathbf{r}$, where

$$\mathbf{r} = \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} \mathbf{R}.$$

The z-component of $\mathbf{r}$ is clearly equal to one, that is $\mathbf{r} \cdot \hat{\mathbf{z}} = 1$.

2.2. Motion Field and Optical Flow

The *motion field* is the vector field induced in the image plane by the relative motion of the observer with respect to the environment. The *optical flow* is the apparent motion of brightness patterns. Under favourable circumstances the optical flow is identical to the motion field (moving shadows or uniform objects in motion could create discrepancies between the motion field and the optical flow. Here, we assume that the motion flow field and the optical flow are the same). The velocity of the image $\mathbf{r}$ of a point $\mathbf{R}$ is given by

$$\frac{d\mathbf{r}}{dt} = \frac{d}{dt} \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} \mathbf{R}.$$

For convenience, we introduce the notation $\mathbf{r}_t$ and $\mathbf{R}_t$ for the time derivatives of $\mathbf{r}$ and $\mathbf{R}$, respectively. We then have

$$\mathbf{r}_t = \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} \mathbf{R}_t - \frac{1}{(\mathbf{R} \cdot \hat{\mathbf{z}})^2} (\mathbf{R}_t \cdot \hat{\mathbf{z}}) \mathbf{R},$$

which can also be written in the compact form

$$\mathbf{r}_t = \frac{1}{(\mathbf{R} \cdot \hat{\mathbf{z}})^2} (\hat{\mathbf{z}} \times (\mathbf{R}_t \times \mathbf{R})).$$

since $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{c} \cdot \mathbf{a})\mathbf{b} - (\mathbf{a} \cdot \mathbf{b})\mathbf{c}$. The vector $\mathbf{r}_t$ lies in the image plane, and so $(\mathbf{r}_t \cdot \hat{\mathbf{z}}) = 0$. Further, $\mathbf{r}_t = \mathbf{0}$ if $\mathbf{R}_t \parallel \mathbf{R}$, as expected.

Finally, noting that $\mathbf{R} = (\mathbf{R} \cdot \hat{\mathbf{z}})\mathbf{r}$, we get

$$\mathbf{r}_t = \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} (\hat{\mathbf{z}} \times (\mathbf{R}_t \times \mathbf{r})).$$

2.3. Rigid Body Motion

In the case of the observer moving relative to a rigid environment with translational velocity $\mathbf{t}$ and rotational velocity ω, we find that the motion of a point in the environment relative to the observer is given by

$$\mathbf{R}_t = -\omega \times \mathbf{R} - \mathbf{t}.$$

Since $\mathbf{R} = (\mathbf{R} \cdot \hat{\mathbf{z}})\mathbf{r}$, we can write this as

$$\mathbf{R}_t = -(\mathbf{R} \cdot \hat{\mathbf{z}})\omega \times \mathbf{r} - \mathbf{t}.$$

Substituting for $\mathbf{R}_t$ in the formula derived above for $\mathbf{r}_t$, we obtain

$$\mathbf{r}_t = -\Big(\hat{\mathbf{z}} \times (\mathbf{r} \times (\mathbf{r} \times \omega - \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} \mathbf{t}))\Big).$$

It is important to remember that there is an inherent ambiguity here, since the same motion field results when distance and the translational velocity are multiplied by an arbitrary constant. This can be seen easily from the above equation since the same image plane velocity is obtained if one multiplies both $\mathbf{R}$ and $\mathbf{t}$ by some constant.

2.4. Brightness Change Equation

The brightness of the image of a particular patch of a surface depends on many factors. It may for example vary with the orientation of the patch. In many cases, however, it remains at least approximately constant as the surface moves in the environment. If we assume that the image brightness of a patch remains constant, we have

$$\frac{dE}{dt} = 0,$$

or

$$\frac{\partial E}{\partial \mathbf{r}} \cdot \frac{d\mathbf{r}}{dt} + \frac{\partial E}{\partial t} = 0,$$

where $\partial E/\partial \mathbf{r} = (\partial E/\partial x, \partial E/\partial y, 0)^T$ is the image brightness gradient. It is convenient to use the notation $E_{\mathbf{r}}$ for this quantity and E_t for the time derivative of the brightness. Then we can write the brightness change equation in the simple form

$$E_{\mathbf{r}} \cdot \mathbf{r}_t + E_t = 0.$$

Substituting for $\mathbf{r}_t$ we get

$$E_t - E_{\mathbf{r}} \cdot \Big(\hat{\mathbf{z}} \times (\mathbf{r} \times (\mathbf{r} \times \omega - \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} \mathbf{t}))\Big) = 0.$$

Now

$$E_{\mathbf{r}} \cdot (\hat{\mathbf{z}} \times (\mathbf{r} \times \mathbf{t})) = (E_{\mathbf{r}} \times \hat{\mathbf{z}}) \cdot (\mathbf{r} \times \mathbf{t}) = ((E_{\mathbf{r}} \times \hat{\mathbf{z}}) \times \mathbf{r}) \cdot \mathbf{t},$$

and by similar reasoning

$$E_{\mathbf{r}} \cdot \Big(\hat{\mathbf{z}} \times (\mathbf{r} \times (\mathbf{r} \times \omega))\Big) = \Big(((E_{\mathbf{r}} \times \hat{\mathbf{z}}) \times \mathbf{r}) \times \mathbf{r}\Big) \cdot \omega,$$

so we have

$$E_t - \Big(((E_{\mathbf{r}} \times \hat{\mathbf{z}}) \times \mathbf{r}) \times \mathbf{r}\Big) \cdot \omega + \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} ((E_{\mathbf{r}} \times \hat{\mathbf{z}}) \times \mathbf{r}) \cdot \mathbf{t} = 0.$$

To make this constraint equation more compact, let us define $c = E_t$, $\mathbf{s} = (E_r \times \hat{\mathbf{i}}) \times \mathbf{r}$, and $\mathbf{v} = -\mathbf{s} \times \mathbf{r}$; then, finally,

$$c + \mathbf{v} \cdot \omega + \frac{1}{\mathbf{R} \cdot \hat{\mathbf{i}}} \mathbf{s} \cdot \mathbf{t} = 0.$$

This is the brightness change equation in the case of rigid body motion.

2.5. Planar Surface

A particularly impoverished scene is one consisting of a single planar surface. The equation for such a surface is

$$\mathbf{R} \cdot \mathbf{n} = 1,$$

where $\mathbf{n}/|\mathbf{n}|$ is a unit normal to the plane, and $1/|\mathbf{n}|$ is the perpendicular distance of the plane from the origin. Since $\mathbf{R} = (\mathbf{R} \cdot \hat{\mathbf{i}})\mathbf{r}$, we can write this as

$$\mathbf{r} \cdot \mathbf{n} = \frac{1}{\mathbf{R} \cdot \hat{\mathbf{i}}},$$

so the constraint equation becomes

$$c + \mathbf{v} \cdot \omega + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) = 0.$$

This is the brightness change equation for a planar surface. Note again the inherent ambiguity in the constraint equation. It is satisfied equally well by two planes with the same orientation but at different distances provided that the translational velocities are in the same proportions.

2.6. Essential Parameters for Planar Surfaces

The brightness change equation can be written as:

$$c + (\mathbf{r} \times \mathbf{s}) \cdot \omega + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) = 0.$$

Now, using the identity $(\mathbf{r} \times \mathbf{s}) \cdot \omega = (\mathbf{s} \times \omega) \cdot \mathbf{r}$, we get:

$$c + (\mathbf{s} \times \omega) \cdot \mathbf{r} + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) = 0.$$

Let us define:

$$\mathbf{W} = \begin{pmatrix} 0 & -\omega_3 & +\omega_2 \\ +\omega_3 & 0 & -\omega_1 \\ -\omega_2 & +\omega_1 & 0 \end{pmatrix},$$

then, $(\mathbf{s} \times \omega) = \mathbf{W}\mathbf{s}$, in which case, we arrive at:

$$c + \mathbf{r}^T \mathbf{W}\mathbf{s} + \mathbf{r}^T \mathbf{n}\mathbf{t}^T \mathbf{s} = 0.$$

Finally, after collecting terms:

$$c + \mathbf{r}^T \mathbf{P}\mathbf{s} = 0,$$

where:

$$\mathbf{P} = \begin{pmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & p_9 \end{pmatrix} = \mathbf{W} + \mathbf{n}\mathbf{t}^T.$$

We will refer to p_i, $i = 1, 2, ..., 9$ as the *essential parameters* (in agreement with Tsai and Huang [20]) since these parameters contain all the information about the planar surface orientation and motion parameters.

The above constraint equation is linear in the elements of P. Several such equations, for different image points, can be used to solve for these parameters. We will show how the special structure of P can be exploited to recover the motion and structure parameters very easily.

3. Recovering Essential Parameters

The nine essential parameters satisfy the following constraint equation:

$$c + \mathbf{r}^T \mathbf{P}\mathbf{s} = 0,$$

or in terms of p, a vector of length 9 whose i-th element is p_i:

$$c + \mathbf{a}^T \mathbf{p} = 0,$$

where:

$$\mathbf{a} = (r_1 s_1, r_1 s_2, r_1 s_3, r_2 s_1, r_2 s_2, r_2 s_3, r_3 s_1, r_3 s_2, r_3 s_3)^T.$$

We can compute them using image brightness $E(x, y, t)$, and its spatial and time derivatives, E_r and E_t, over some region I in the image plane. Since there are only eight motion and surface parameters to recover (There are three components of each of ω, t, and n. However, the translational velocity and the surface normal can be recovered up to a scale factor.), only eight of the p_i's are independent. This implies that we can arbitrarily fix one of the essential parameters, and compute the remaining eight using eight independent points (At each point, we get one constraint and we have eight unknowns to recover).

Let $\mathbf{p}' = (p'_1, p'_2, ..., p'_8, 0)$ denote the solution obtained by setting the last element equal to zero. If we define:

$$\tilde{\mathbf{p}}' = (p'_1, p'_2, ..., p'_8),$$

$$\tilde{\mathbf{a}} = (r_1 s_1, r_1 s_2, r_1 s_3, r_2 s_1, r_2 s_2, r_2 s_3, r_3 s_1, r_3 s_2)^T,$$

then the above constraint equation reduces to:

$$\tilde{\mathbf{a}}^T \tilde{\mathbf{p}}' + c = 0.$$

Using eight independent points, we can solve the following linear matrix equation:

$$\mathbf{A}\tilde{\mathbf{p}}' + \mathbf{c} = 0,$$

where:

$$\mathbf{A} = (\tilde{\mathbf{a}}_1, \tilde{\mathbf{a}}_2, ..., \tilde{\mathbf{a}}_8)^T, \quad \mathbf{c} = (c_1, c_2, ..., c_8).$$

The solution of the above equation is:

$$\tilde{\mathbf{p}}' = -\mathbf{A}^{-1}\mathbf{c}.$$

Image brightness values are distorted with sensor noise and quantization error. These inaccuracies are

further accentuated by methods used for estimating the brightness gradient. Thus it is not advisable to base a method on measurements at just a few points. Instead we propose to minimize the error in the brightness constraint equation over the whole region I in the image plane. So we choose the essential parameters that minimize:

$$\iint_I (\tilde{\mathbf{s}}^T \tilde{\mathbf{p}}' + c)^2 \, dx \, dy.$$

The solution, in this case, is given by:

$$\tilde{\mathbf{p}}' = -\left(\iint_I \tilde{\mathbf{a}}\tilde{\mathbf{a}}^T \, dx \, dy\right)^{-1} \left(\iint_I c\tilde{\mathbf{a}} \, dx \, dy\right).$$

Note that, in general, the true p_9 is nonzero. We can show that the solution obtained through the assumption that $p'_9 = 0$, $\mathbf{p}'$, and the true solution (denoted by $\mathbf{p}$) are related by the equation:

$$\mathbf{p} = \mathbf{p}' + p_9\mathbf{u}, \quad \mathbf{u} = (1,0,0,0,1,0,0,0,1)^T.$$

The proof goes as follows. Since $\mathbf{s} = (E_r \times \hat{\mathbf{z}}) \times \mathbf{r}$, then $\mathbf{r}^T\mathbf{s} = \mathbf{r} \cdot ((E_r \times \hat{\mathbf{z}}) \times \mathbf{r}) = 0$. For any arbitrary constant l, such that $\mathbf{L} = lI$ (I is the identity matrix), we have:

$$\mathbf{r}^T\mathbf{L}\mathbf{s} = 0.$$

If we add this to our costraint equation, we get:

$$c + \mathbf{r}^T(\mathbf{W} + \mathbf{n}\mathbf{t}^T + \mathbf{L})\mathbf{s} = 0.$$

It is immediately apparent that any $\mathbf{P}'$ of the form:

$$\mathbf{P}' = \begin{pmatrix} p'_1 & p'_2 & p'_3 \\ p'_4 & p'_5 & p'_6 \\ p'_7 & p'_8 & p'_9 \end{pmatrix} = \mathbf{W} + \mathbf{n}\mathbf{t}^T + \mathbf{L}$$

will also satisfy our constraint equation. Therefore, the two solutions for the essential parameters are related by:

$$\mathbf{P} = \mathbf{P}' - \mathbf{L},$$

or in terms of $\mathbf{p}$ and $\mathbf{p}'$:

$$\mathbf{p} = \mathbf{p}' - l\mathbf{u}, \quad \mathbf{u} = (1,0,0,0,1,0,0,0,1),$$

for some constant l.It follows that $p_9 = p'_9 - l$. Now if $p'_9 = 0$, then $l = -p_9$, and so:

$$\mathbf{p} = \mathbf{p}' + p_9\mathbf{u}.$$

In term of matrices $\mathbf{P}$, and $\mathbf{P}'$:

$$\mathbf{P} = \mathbf{P}' + p_9\mathbf{I}.$$

The procedure for determining p_9, and subsequently $\mathbf{P}$ is presented in the appendix. For now, we assume that $\mathbf{P}$ is known (note that we have, so far, shown how to compute $\mathbf{P}'$ and not $\mathbf{P}$).

4. Recovering Motion and Structure

We now show how to compute the parameters of the translational motion and the plane orientation from the essential parameters. Once these are known, the rotational parameters are determined from:

$$\mathbf{W} = \mathbf{P} - \mathbf{n}\mathbf{t}^T$$

Since $\mathbf{W}$ is skew-symmetric:

$$\mathbf{P}^* = \mathbf{P} + \mathbf{P}^T = (W + W^T) + (\mathbf{t}\mathbf{n}^T + \mathbf{n}\mathbf{t}^T) = (\mathbf{t}\mathbf{n}^T + \mathbf{n}\mathbf{t}^T).$$

We will derive our results in terms of the normalized $\mathbf{n}$,$\mathbf{t}$, and $\mathbf{P}^*$, that is:

$$\tilde{\mathbf{n}} = \frac{\mathbf{n}}{|\mathbf{n}|}, \quad \tilde{\mathbf{t}} = \frac{\mathbf{t}}{|\mathbf{t}|} = \mathbf{t}, \quad \tilde{\mathbf{P}}^* = \frac{1}{|\mathbf{n}|\,|\mathbf{t}|}\mathbf{P}^*.$$

In order to present the solutions for $\tilde{\mathbf{n}}$ and $\tilde{\mathbf{t}}$, it is necessary to express the eigenvalue decomposition of $\tilde{\mathbf{P}}^*$ in terms of these vectors. We will do so in the next lemma.

Lemma 1: Let $\tilde{\mathbf{P}}^* = \mathbf{U}\mathbf{A}\mathbf{U}^T$ be the eigenvalue decomposition of $\tilde{\mathbf{P}}^* = (\tilde{\mathbf{t}}\tilde{\mathbf{n}}^T + \tilde{\mathbf{n}}\tilde{\mathbf{t}}^T)$, and let $r = \frac{1}{2}\mathrm{Trace}\tilde{\mathbf{P}}^*$. Then:

$$\mathbf{A} = \begin{pmatrix} r-1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & r+1 \end{pmatrix},$$

$$\mathbf{U} = \left[\frac{\tilde{\mathbf{t}} - \tilde{\mathbf{n}}}{\sqrt{2(1-r)}} \quad \frac{\tilde{\mathbf{t}} \times \tilde{\mathbf{n}}}{|\tilde{\mathbf{t}} \times \tilde{\mathbf{n}}|} \quad \frac{\tilde{\mathbf{t}} + \tilde{\mathbf{n}}}{\sqrt{2(1+r)}}\right].$$

Proof: $(\tilde{\mathbf{t}} \times \tilde{\mathbf{n}})$ is the eigenvector corresponding to the zero eigenvalue of $\tilde{\mathbf{P}}^*$ since $(\tilde{\mathbf{t}}\tilde{\mathbf{n}}^T + \tilde{\mathbf{n}}\tilde{\mathbf{t}}^T)\tilde{\mathbf{t}} \times \tilde{\mathbf{n}} = \tilde{\mathbf{t}}\big(\tilde{\mathbf{n}} \cdot (\tilde{\mathbf{t}} \times \tilde{\mathbf{n}})\big) + \tilde{\mathbf{n}}\big(\tilde{\mathbf{t}} \cdot (\tilde{\mathbf{t}} \times \tilde{\mathbf{n}})\big) = 0$. Since it is symmetric, $\mathbf{P}^*$ has 3 orthogonal eigenvectors. The other two eigenvectors are, therefore, in the plane containing $\tilde{\mathbf{t}}$ and $\tilde{\mathbf{n}}$. Let $\mathbf{u} = \alpha\tilde{\mathbf{t}} + \beta\tilde{\mathbf{n}}$ and λ denote an eigenvector-eigenvalue pair for some α and β (to be determined). Then

$$(\tilde{\mathbf{t}}\tilde{\mathbf{n}}^T + \tilde{\mathbf{n}}\tilde{\mathbf{t}}^T)(\alpha\tilde{\mathbf{t}} + \beta\tilde{\mathbf{n}}) = \lambda(\alpha\tilde{\mathbf{t}} + \beta\tilde{\mathbf{n}}),$$

which simplifies to

$$[\alpha(\tilde{\mathbf{t}} \cdot \tilde{\mathbf{n}}) + \beta(\tilde{\mathbf{n}} \cdot \tilde{\mathbf{n}})]\tilde{\mathbf{t}} + [\alpha(\tilde{\mathbf{t}} \cdot \tilde{\mathbf{t}}) + \beta(\tilde{\mathbf{t}} \cdot \tilde{\mathbf{n}})]\tilde{\mathbf{n}} = \lambda\alpha\tilde{\mathbf{t}} + \lambda\beta\tilde{\mathbf{n}}.$$

Since $(\tilde{\mathbf{t}} \cdot \tilde{\mathbf{n}}) = r$, we have:

$$\begin{pmatrix} r-\lambda & 1 \\ 1 & r-\lambda \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = 0.$$

The solutions for λ are given by:

$$\lambda = r \pm 1.$$

Substituting for λ into the earlier equations, we get:

$$\alpha = \pm\beta.$$

Using these into the equation for $\mathbf{u}$, and normalizing the results yield:

$$\mathbf{u} = \frac{\tilde{\mathbf{t}} + \tilde{\mathbf{n}}}{\sqrt{2(1+r)}}.$$

Note that since $|r| < 1$, we have $\lambda_1 < \lambda_2 = 0 < \lambda_3$. ∎

We can now determine $\tilde{\mathbf{t}}$ and $\tilde{\mathbf{n}}$. Let $\mathbf{u}_i$ denote the i-th column of $\mathbf{U}$. From lemma 1:

$$\mathbf{u}_1 = \frac{\tilde{\mathbf{t}} - \tilde{\mathbf{n}}}{\sqrt{2(1-r)}}, \quad \mathbf{u}_3 = \frac{\tilde{\mathbf{t}} + \tilde{\mathbf{n}}}{\sqrt{2(1+r)}}.$$

From the expressions for the eigenvectors, it follows that:

$$\tilde{\mathbf{t}} + \tilde{\mathbf{n}} = \sqrt{2(1+r)}\mathbf{u}_3. \quad \tilde{\mathbf{t}} - \tilde{\mathbf{n}} = \sqrt{2(1-r)}\mathbf{u}_1.$$

Solving these for $\tilde{\mathbf{t}}$ and $\tilde{\mathbf{n}}$, we get:

$$\tilde{\mathbf{n}} = \sqrt{\frac{1+r}{2}}\mathbf{u}_3 - \sqrt{\frac{1-r}{2}}\mathbf{u}_1, \quad \tilde{\mathbf{t}} = \sqrt{\frac{1+r}{2}}\mathbf{u}_3 + \sqrt{\frac{1-r}{2}}\mathbf{u}_1.$$

Since the choice of the signs of the eigenvectors are arbitrary, we should repeat the above procedure with the sign of either $\mathbf{u}_1$ or $\mathbf{u}_3$ reversed:

$$\mathbf{u}_1 = -\frac{\tilde{\mathbf{t}} - \tilde{\mathbf{n}}}{\sqrt{2(1-r)}}, \quad \mathbf{u}_3 = \frac{\tilde{\mathbf{t}} + \tilde{\mathbf{n}}}{\sqrt{2(1+r)}}.$$

In this case, the second solution is obtained from the following equations:

$$\tilde{\mathbf{n}} = \sqrt{\frac{1+r}{2}}\mathbf{u}_3 + \sqrt{\frac{1-r}{2}}\mathbf{u}_1, \quad \tilde{\mathbf{t}} = \sqrt{\frac{1+r}{2}}\mathbf{u}_3 - \sqrt{\frac{1-r}{2}}\mathbf{u}_1.$$

This is the dual solution for the planar surfaces.

The special case of $\tilde{\mathbf{t}} \parallel \tilde{\mathbf{n}}$ corresponds to when the matrix $\tilde{\mathbf{P}}^*$ has multiple eigenvalues. Then, either:

1) $r = 1$, for which $\lambda_1 = \lambda_2 = 0$. Then the two solutions merge to the single one:

$$\tilde{\mathbf{n}} = \tilde{\mathbf{t}} = \mathbf{u}_3,$$

or:

2) $r = -1$, so that $\lambda_2 = \lambda_3 = 0$. Here, the unique solution is given by:

$$\tilde{\mathbf{n}} = -\tilde{\mathbf{t}} = \mathbf{u}_1.$$

Since the translational vector and the surface normal can be recovered up to a scale factor, we can, without loss of generality, set $|\mathbf{t}| = 1$. It can be easily shown that:

$$\lambda(\mathbf{P}^*) = |\mathbf{n}||\mathbf{t}|\lambda(\tilde{\mathbf{P}}^*) = |\mathbf{n}|\lambda(\tilde{\mathbf{P}}^*).$$

Therefore:

$$\lambda_3(\mathbf{P}^*) - \lambda_1(\mathbf{P}^*) = 2|\mathbf{n}|,$$

or

$$|\mathbf{n}| = \frac{1}{2}(\lambda_3(\mathbf{P}^*) - \lambda_1(\mathbf{P}^*)).$$

Since

$$\mathbf{P}^* = |\mathbf{n}||\mathbf{t}|\tilde{\mathbf{P}}^* = |\mathbf{n}|\tilde{\mathbf{P}}^*,$$

we use the following equation to normalize $\mathbf{P}^*$ in the first place:

$$\tilde{\mathbf{P}}^* = \frac{2}{\lambda_3(\mathbf{P}^*) - \lambda_1(\mathbf{P}^*)}\mathbf{P}^*.$$

Once we compute $\tilde{\mathbf{n}}$ and $\tilde{\mathbf{t}}$ from the equations given earlier, we can determine $\mathbf{n}$ and $\mathbf{t}$ through proper scaling:

$$\mathbf{n} = |\mathbf{n}|\tilde{\mathbf{n}}, \quad \mathbf{t} = \tilde{\mathbf{t}}.$$

We then solve for the rotation parameters by substituting the solutions for $\mathbf{n}$ and $\mathbf{t}$ into the equation:

$$\mathbf{W} = \mathbf{P} - \mathbf{n}\mathbf{t}^T.$$

Even though we gave a complete and compact proof of the dual solution in an earlier paper[15], it is intriguing to confirm those results with our closed form solution. We showed that the two solutions are related by:

$$\mathbf{n}^* = |\mathbf{n}|\mathbf{t}, \quad \mathbf{t}^* = \frac{\mathbf{n}}{|\mathbf{n}|}, \quad \omega^* = \omega + \mathbf{n} \times \mathbf{t},$$

where we have arbitrarily set $|\mathbf{t}| = 1$. The two solutions given earlier for $\mathbf{n}$ and $\mathbf{t}$ already satisfy the duality relationship given above. It remains to show the same for the rotation parameters. We only have to show that:

$$\mathbf{W}^* + \mathbf{n}^*\mathbf{t}^{*T} = \mathbf{P},$$

where

$$\mathbf{W}^* = \mathbf{W} + \begin{pmatrix} 0 & n_1t_2 - n_2t_1 & n_1t_3 - n_3t_1 \\ n_2t_1 - n_1t_2 & 0 & n_2t_3 - n_3t_2 \\ n_3t_1 - n_1t_3 & n_3t_2 - n_2t_3 & 0 \end{pmatrix},$$

or

$$\mathbf{W}^* = \mathbf{W} + \mathbf{n}\mathbf{t}^T - \mathbf{t}\mathbf{n}^T.$$

Substituting for $\mathbf{n}^*$, $\mathbf{t}^*$, and $\mathbf{W}^*$ into the earlier equation, and simplifying the results, we get:

$$\mathbf{W} + \mathbf{n}\mathbf{t}^T = \mathbf{P}.$$

5. Summary

In this paper, we presented a closed form solution for recovering the motion of an observer with respect to a planar surface without having to compute the optical flow as an intermediate step. We only need the image intensity gradients at a minimum of 8 points. However, in general, it is better to compute gradients in a larger portion of the image to reduce the noise effects. We first employed a constraint equation we developed for planar surfaces to compute 9 intermediate parameters, the elements of a 3x3 matrix. We referred to them as *essential parameters*. The special structure of this matrix allows us to compute the motion and plane parameters very easily.

Appendix

In the previous sections, we showed how the motion paprameters can be recovered once the essential paprameters are known. However, the brightness change constraint equation allowed us to determine the matrix $\mathbf{P}'$ (a particular solution of $\mathbf{P}$ with the last element set to zero). We showed that the two solution are related by:

$$\mathbf{P} = \mathbf{P}' + p_9\mathbf{I}.$$

Here, we show how to determine p_9, and consequently $\mathbf{P}$. For simplicity, let $p_9 = l$, and let $\mathbf{P} = \mathbf{UAV}^T$ denote the eigenvalue decomposition of $\mathbf{P}$, where $(\mathbf{A} = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3))$, then:

$$\mathbf{P}' = \mathbf{UAV}^T - l\mathbf{I} = \mathbf{UAV}^T - l\mathbf{UV}^T.$$

If $\mathbf{L} = l\mathbf{I}$ then:

$$\mathbf{P}' = \mathbf{UAV}^T - \mathbf{ULV}^T = \mathbf{U}(\mathbf{A} - \mathbf{L})\mathbf{V}^T.$$

Similarly, if $\mathbf{P}^* = \mathbf{P} + \mathbf{P}^T = \mathbf{UAU}^T$ denotes the eigenvalue decomposition of $\mathbf{P}^*$, then:

$$\mathbf{P}'^* = \mathbf{P}' + \mathbf{P}'^T = \mathbf{P} + \mathbf{P}^T - 2\mathbf{L} = \mathbf{P}^* - 2\mathbf{L} = \mathbf{U}(\mathbf{A} - 2\mathbf{L})\mathbf{U}^T.$$

In lemma 1, we showed that $\lambda_1 < \lambda_2 = 0 < \lambda_3$ (and when $\mathbf{t} \parallel \mathbf{n}$, we get two zero eigenvalues). Therefore, the eigenvalues of $\mathbf{P}'^*$ can be arranged in the form:

$$\lambda_1 - 2l < -2l < \lambda_3 - 2l.$$

It follows that $l = -\frac{1}{2}\lambda_2(\mathbf{P}')$.

So in summary, we assume that $p_9 = 0$, and solve for the essential parameters (elements of $\mathbf{P}'$). The eigenvalue decomposition of $\mathbf{P}'^*$ allows us to determine the unknown shift ($p_9 = -\frac{1}{2}\lambda_2$), and then, $\mathbf{P}$ from:

$$\mathbf{P} = \mathbf{P}' - \frac{1}{2}\lambda_2(\mathbf{P}')\mathbf{I}$$

Finally, the solution of the motion and structure parameters are determined from the equations given earlier in terms of the trace and eigenvectors of $\mathbf{P}^*$ (note that the eigenvectors of $\mathbf{P}^*$ and $\mathbf{P}'^*$ are the same).

6. References

[1] Adiv, G., "Determining 3-D Motion and Structure from Optical Flow Generated by Several Moving Objects," COINS TR 84-07, Computer and Information Science, University of Massachusetts, Amherst, MA, April 1984.

[2] Aloimonos, J., Chou, P. B., "Detection of Surface Orientation and Motion from Texture: 1. The Case of Planes," TR 161, Department of Computer Science, Univ. of Rochester, Rochester, NY, January 1985.

[3] Ballard, D.H., Kimball, O.A., "Rigid Body Motion from Depth and Optical Flow," TR 70, Computer Science Department, Univ. of Rochester, Rochester, NY, 1981.

[4] Bruss, A.R., Horn, B.K.P., "Passive Navigation," *Computer Vision, Graphics, and Image Processing*, Vol. 21, pp. 3–20, 1983.

[5] Buxton, B.F., et al., "3D Solution to the Aperature problem," *Proceedings of the Sixth European Conference on Artificial Intelligence*. pp. 631–640, September 1984.

[6] Fennema, C.L., Thompson W.B., "Velocity Determination in Scenes Containing Several Moving Objects," *Computer Graphics and Image Processing*, 9, pp. 301–315, 1979.

[7] Hildreth, E.C., *The Measurement of Visual Motion*, MIT Press, 1983.

[8] Horn, B.K.P., Schunck, B.G., "Determining Optical Flow," *Artificial Intelligence*, Vol. 17, pp. 185–203, 1981.

[9] Kanatani, K., "Detecting the Motion of a Planar Surface by Line and Surface Integrals," *Computer Vision, Graphics, and Image Processing*, 29, pp. 13–22, 1985.

[10] Longuet-Higgins, H.C., Prazdny, K., "The Interpretation of a Moving Retinal Image," *Proc. of Royal Society of London*, Series B, Vol. 208, pp. 385–397, 1980.

[11] Longuet-Higgins, H.C., "A Computer Algorithm for Reconstructing a Scene from Two Projections," *Nature*, Vol. 293, pp. 131–133, 1981.

[12] Longuet-Higgins, H.C., "The Visual Ambiguity of a Moving Plane," *Proc. of the Royal Society of London*, B 223, pp. 165–175, 1984.

[13] Nagel, H., "On the Derivation of 3D Rigid Point Configurations from Image Sequences," *Proceedings of Pattern Recognition and Image Processing*, Dallas, Texas, 1981.

[14] Negahdaripour, S., Horn, B.K.P., "Determining 3-D Motion of Planar Objects from Brightness Patterns," *Proceedings of Ninth Int. Joint Conf. on A.I.*, pp. 898–901, 1985.

[15] Negahdaripour, S., Horn, B.K.P., "Direct Passive Navigation," to appear in *IEEE Trans. Pattern Analysis and Machine Intelligence*.

[16] Roach, J.W., Aggarwal, J.K., "Determinig the Movement of Objects from a Sequence of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, November 1980.

[17] Subbarao, M., Waxman, A.M., "On the Uniqueness of Image Flow Solutions for Planar Surfaces in Motion," *Proc. of the Third Workshop on Computer Vision: Representation and Control*, pp. 129-140, 1985.

[18] Sugihara, K., Sugie, N., "Recovery of Rigid Structure from Orthographically Projected Optical Flow," TR 8304, Dep. of Inf. Science, Nagoya University, Nagoya, Japan, October 1983.

[19] Tsai, R.Y., Huang, T.S., Zhu, W.L., "Estimating 3-D Motion Parameters of a Rigid Planar Patch, II: Singular Value Decomposition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-30, No. 4, August 1982.

[20] Tsai, R.Y., Huang, T.S., "Uniqueness and Estimation of 3-D Motion Parameters of Rigid Objects with Curved Surfaces," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 1, January 1984.

[21] Waxman, A.M., Ullman, S., "Surface Structure and 3-D Motion from Image Flow: A Kinematic Analysis," CAR-TR-24, Comp. Vision Laboratory, Center for Automation Research, University of Maryland, College Park, MD, October 1983.

[22] Waxman, A.M., Wohn, K. "Contour Evolution, Neighborhood Deformation and Global Image Flow: Planar Surfaces in Motion," CAR-TR-58, Comp. Vision Laboratory, Center for Automation Research, Univ. of Maryland, College Park, MD, April 1984.

GC976145-42

ANALYSIS OF AN ALGORITHM FOR DETECTION OF TRANSLATIONAL MOTION

Igor Pavlin, Edward Riseman and Allen Hanson

Computer and Information Sciences Department
University of Massachusetts, Amherst, MA 01003

ABSTRACT

This report presents an extensive testing of an algorithm for the recovery of translational motion parameters of a sensor moving through a static environment. The algorithm has been evaluated using synthetic images in terms of the number of feature points matched between frames, the relative angle between camera orientation and direction of translation, uncorrelated and correlated noise, and computational cost.

The algorithm appears to be robust across a very wide range of camera translations, using only as few as 8 feature points. When the angle between the direction of translation and the direction of view is in certain range of angles the algorithm experiences difficulties.

In addition, an improvement in the speed and possibly the accuracy of the search is suggested. By the reasonable assumption of smoothness in the error surface, many stages of iterative search may be avoided.

1. INTRODUCTION

This paper describes results of an extensive testing of the algorithm developed by D. Lawton [LAW84] for recovery of translational motion parameters. This algorithm avoids the computation of an image displacement field prior to recovery of the parameters for the axis of translational motion. The axis of translation may also be specified by the point where it intersects the image plane, called the focus of expansion/contraction (abbreviated to FOE/C).

The global nature of the technique arises from the use of many "interesting" or distinguishable points spread over an image. Despite problems of noise, false feature matches, occlusion of features, and other causes of unreliable information in some parts of the image, the solutions obtained can be quite stable. The robustness and accuracy are a consequence of the algorithm's global and local characteristics: a search for the minimum of the global error associated with a set of points whose motion is jointly constrained, and local correlation measurements to find the best match of each point contributing to the global error. Intuitively, the global nature of the search for the correct position of the FOE/C is manifested in the many "votes" of different displacement vectors for the position of the FOE/C, and consequently the algorithm demonstrates a high degree of robustness and accuracy.

This paper consideres only the case of translational motion. However, there are two closely related cases which we believe will yield to a similar approach and results. For example, translation of the camera constrained to a known plane, with simultaneous rotation about the axis perpendicular to that plane, is a two-dimensional problem. Similarly, pure sensor rotation about a fixed axis is described with only three parameters (two of them specifying the axis of rotation and one the extent of the rotation). We did not consider these cases of constrained motion in this paper, but we believe their robustness will be similar, because the same type of global constraints on the local feature matches is available.

Evaluation of performance has been carried out be examining the influence of the following parameters:

1. number of feature points in the image plane which are matched between two frames.
2. the accuracy with which the direction of translation can be recovered as a function of the relative orientation of the camera line of sight to the direction of motion.
3. resolution of the sampling of the FOE/C during the search.
4. size of the window used for feature correlation between frames.
5. resolution of the correlation matching for each feature (i.e., step size of feature displacements for interpolation and matching).
6. the efficiency of computation, and the robustness of the method with respect to the correlated and uncorrelated noise.

The motion sequence used in our experiments is created using a computer graphics system which incorporates

This work was supported by the Defense Advanced Research Projects Agency, DARPA under grant N00014-82-K-0464

ray-tracing techniques [WHI80]. The sequence is a reasonable substitute for real-world images since light is handled somewhat realistically (shadows, specular reflection, decrease of the light intensity with the distance, and the physical laws of reflection and refraction are incorporated). The advantage of synthetic images is that they allow the experimental situation to be controlled with an accurate model of the camera motion in an environment. Evaluation of the actual performance of the algorithm on real world images will be performed as data bases of motion sequences become available.

As we have mentioned, one of the issues associated with the algorithm that we have addressed in this paper is the efficiency of the search for the FOE/C, which involves a global (sparse) sampling of the error surface followed by a finer-resolution local hill-climbing search for the global minimum of the error surface. There may be unnecessary inefficiencies in this search algorithm. Also, when the error surface around the minimum becomes relatively flat and exhibits local small fluctuations, the local hill-climbing technique might fail.

We suggest an improvement of the algorithm by introducing a smoothness constraint on the error surface. This approach is based upon strong intuitions about why the error surface should be smooth if more than a few points are tracked between a pair of frames. The global assumption of smoothness allows the use of a surface fitting algorithm to speed up the search for the FOE/C, while not causing the performance to deteriorate significantly. The approach may also allow the search for the minimum to be more reliable by eliminating the local hill-climbing technique.

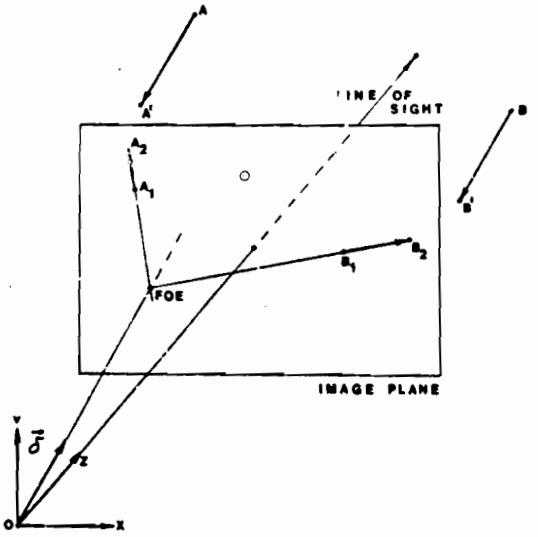
2. BACKGROUND

In this section we present a brief review of the Lawton algorithm and the concepts concerning translational motion that are necessary to follow the discussion of the experiments in the following section. The reader is encouraged to consult [LAW83, LAW84] for more details.

2.1 Translational Motion and Displacement Fields

A displacement field is defined as a vector field produced by the changes in the position of the images of environmental points over time [GIB50]. In the case of pure translational motion of the camera in a static environment, the intersection of the image plane and the vector describing the translational motion of the focal point of the camera is called the FOE/C. The displacement vector of any image feature lies on the radial line connecting the FOE/C to the feature, see Figure 1.

As few as two image points would be sufficient to identify the correct translational motion. If the displacement of two points that are not collinear could be accurately tracked from one frame to another, their intersection in the image would determine the FOE/C, which in turn unambiguously specifies the direction of translation.



The camera model and the displacement field produced by two environmental points A and B during the motion of the camera specified by the translational vector $\vec{\delta}$.

Figure 1: Camera model and FOE/C.

In practice, however, establishing the proper correspondence between the parts of images in two or more consecutive frames is difficult. Correspondence between similar parts of two images is established using techniques such as a scalar valued correlation function or symbolic (token) matching of key image events. However, there are a variety of problems that make such mechanisms unreliable: the size of the local area to be searched, presence of noise in the image, occlusion of surfaces that previously were visible, insufficient resolution of the image, etc.

Redundancy and global constraints on the feature dynamics across frames can be used to overcome these difficulties. Since the displacement vectors of all features in a static environment are constrained to emanate from the FOE/C, the use of additional features should more accurately constrain the correct position of the FOE/C. Use of redundant features would compensate for the "noise" introduced by those features that provide weak or incorrect information (e.g., features in low contrast or homogeneous regions, features at occlusion boundaries). However, the use of a large number of features is undesirable from the point of view of computational efficiency.

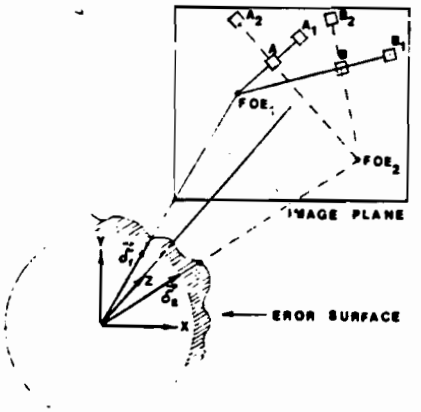
2.2 Error function for FOE/C search

The position of the FOE/C can be obtained in two basic steps. The first step is the extraction of features, and

the second step is the search for the position of the FOE/C. The whole search procedure is symbolised in Figure 2.

The feature extraction process is responsible for the extraction of distinguishing points which can be tracked from one frame to another. Contour points of high curvature are a good choice because they are less likely to produce ambiguous matches in succeeding frames. The contours can be extracted using a variety of techniques: thresholding, zero-crossings, boundary curvature measures, local contrast measurements, sharpness of the autocorrelation function, etc.

The direction of translation is then found by a search across hypothesised FOE/C positions in the image plane. For each hypothesised position of the FOE/C and for each feature in the image the best correlation between frames is found along the radial line connecting the hypothesised



The position of the FOE/C in the image plane is hypothesized. The FOE/C itself is at the intersection of the image plane and the translation vector $\vec{\delta}$. The position of the FOE/C also uniquely determines a point on the unit sphere where the translation vector $\vec{\delta}$ pierces the unit sphere. At that point, the error associated with the mismatch of the features A, A_1 and B, B_1 is calculated. For a different FOE/C the best match for features A and B might be found along a radial path at the positions A_2 and B_2. The error at that FOE/C is expected to be almost always higher when the position of the assumed FOE/C is significantly incorrect. This process is repeated for many hypothesized positions of the FOE/C resulting in the error surface on the unit sphere. The position of the minimum of the error surface determines the selection for the correct axis of translation (or, equivalently, the correct FOE/C).

Figure 2: Search for the FOE/C.

FOE/C and the feature. If a feature, say A, has moved in the subsequent image to the position A_1 some distance d along the radial line, then a subimage around A is expected to correlate nearly perfectly exactly d units along the radial line in the next frame. An equivalent representation of the information is given as an error measure between features A and A_1:

$$error_{feature_A} = 1 - corr(feature_A, feature_{A_1}).$$

Several measures were examined by Lawton for feature matching: the normalised correlation, the Moravec correlation (which was used in this paper) [MOR77], and the normalised absolute value difference. These correlation functions differ in terms of speed and precision. The speed increases and the accuracy decreases in the order listed above. The size of the n x n correlation window can affect accuracy (in this paper the size is varied from 3 to 11). In addition, one must consider the alternative forms of interpolation, because subimages from one frame are displaced to positions in the next frame that are not aligned with pixels; here we use bilinear interpolation.

As an alternative, we are currently examining the use of symbolic features, or "tokens", i.e., interesting points with a set of attribute-value pairs, for the use in symbolic matching. The replacement of correlation matching with symbolic matching has the potential of significantly increasing the speed of computation[1], but it was not used here.

If the position of the hypothesised FOE/C is very close to the correct FOE/C then most of the features will return good matches and small errors, resulting in a very small total error for that particular position of the FOE/C. If the position of the hypothesised FOE/C is far from the correct position, the computed total error will be significant, since many features will return a poor match. Proceeding in this fashion an error surface can be constructed over the image plane, with the minimal error expected at the position of the correct FOE/C. To achieve a more uniform sampling of the hypothesised positions of the FOE/C, it is much more convenient to represent the orientation of the axis of translation and associated error values on the unit sphere; this is accomplished by projecting error surface values onto the unit sphere centered at the camera focus.

The error surface on the unit sphere can be constructed starting with a coarse sampling over polar angles. In the experiments that follow, the hypothesised position of the FOE/C on the unit sphere is specified by the angles α and β, where α is the angle between the projection of the translational axis on the Y-Z plane and the z-axis, and β is the angle between the projection of the translational axis on the X-Z plane and the z-axis. The resolution of the sampling is one factor determining the precision of the method and speed of the search. The grid spacing for sampling in the (α, β) coordinate system was roughly 45, 22.5, and 11.25 degrees in our experiments.

Once the coarse sampled error surface is found, the

[1] Lawton, private communication

search for the minimum of the error surface is continued locally at the global minimum of the coarse sampling. The smallest error value of the coarse global search is taken as the initial guess for a finer resolution local search. A simple hill-climbing technique is utilised under the assumption that the error surface is monotonically decreasing in the vicinity of the global minimum. Error values for FOE/C at eight new neighborhood points around the current minimum are computed. The new neighborhood points were chosen at half the distance from the previous neighbors. The point with the smallest error among these nine points is found and becomes the new starting point for the next iteration of the local search. The new neighborhood is searched, again with a radius equal to the one half of the previous spacing between the neighbors. The procedure is repeated until the radius of the neighborhood becomes smaller than a limiting value δ_m. If the error surface near the minimum is convex, this guarantees that, when the local search is completed, the position of the minimum of the error surface will be found with an accuracy of $\pm\delta_m$ in the polar coordinates. The purpose of the global search was to localise the search for the minimum of the error surface in such a convex neighborhood.

3. EXPERIMENTS

In this section we describe the experiments performed using the translational motion algorithm explained in the previous section. This is followed by a discussion of the results (Section 3.2) and suggestions for possible improvements (Section 4). Experiments are done using the VISIONS image operating system [HAN84].

3.1 Description of experiments

The initial analysis of the algorithm was performed on the same natural outdoor images (the road sign images) described by [WIL80, LAW84]; these images have a spatial resolution of 128 x 128 pixels. We repeated these experiments with only a slight discrepancy from the values determined by the previous research, probably due to the use of different features. The results were satisfactory even with only four feature points, and the use of 3 feature points seems to be as effective as 64 feature points. The global search was performed with a resolution of 22.5°, and the local search parameter was $\delta_m = 0.05$ radians. The total number of FOE/C positions explored was around 160 in each of the experiments.

The symbol Δ in the table stands for difference (in percent) between experiments indicated in parenthesis. In the road-sign images the exact translational axis is not known. Therefore, the absolute precision of the search is unknown, and it is not clear whether there is any systematic deviation from the correct translational axis, nor whether the algorithm would perform well for other viewpoints relative to the direction of motion.

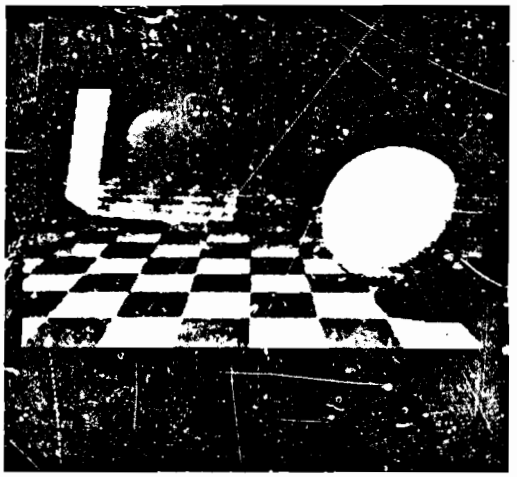
The Road-Sign Sequence

Exper. No.	No. of points	tranal. x coord.	tranal. y coord.	tranal. z coord.	time min.
1	4	-0.8451	-0.4353	0.3101	7
2	8	-0.8338	-0.4150	0.3638	10
3	16	-0.8326	-0.4285	0.3736	17
4	32	-0.8439	-0.4218	0.3313	34
5	64	-0.8439	-0.4218	0.3313	68
6*	140	-0.8373	-0.4204	0.3493	?
$\Delta(2,6)$		-1.19%	-2.38%	2.86%	
$\Delta(5,6)$		-0.79%	-0.33%	5.15%	

* (Lawton)

To assure a controlled experimental situation, synthetic images were constructed using computer graphics techniques. One of the images is shown in Figure 3.

The images have a resolution of 256 x 256 pixels. In these experiments the camera position and motion is known exactly. The model of lighting includes shadows, reflective



A synthetic image in which positions of all elements are controllable.

Figure 3: Synthetic image used in experiments.

and transmissive surfaces, specular reflection, and decreasing light intensity as a function of distance. Feature points on this image appeared at some of the corners of checkerboard squares, on the prism, and on the sphere boundaries. The important camera parameters for this simulated environment are the focal length ($f = 5$ - arbitrary units), the width ($w = 4$) and the height ($h = 4$) of the image plane. This corresponds to a field of view of $43.6° \approx 45°$ as opposed to 90° in Lawton's experiments. The line of sight is inclined 15° towards the horizontal checkerboard. The center of the 10 x 10 checkerboard is at $y = -2$, $z = 7$. In

all experiments the displacement of the camera was calculated so that the maximal displacement of the nearest point on the checkerboard was about 7 pixels and the maximal displacement of the furthest point on the checkerboard was about 4 pixels.

First, a set of translations in the Y-Z plane is generated, starting from the line of sight (z-axis), and increasing α by increments of 15° from 0° to 90°, and then by increments of 30° to 180° (keeping $\beta = 0$). The translations parallel to the image plane in the directions along the x-axis and the bisector of the first X-Y quadrant are also analysed, as well as the translation along an "arbitrary" direction specified by the angles $(\alpha, \beta) = (60°, 30°)$.

Parameters are varied from a default set of parameters. The defaults are: correlation window displacements of 1 pixel (with a maximum allowed displacement of 10 pixels), window size of 7x7, sampling in polar coordinates of roughly every 45°, and local search precision of $\delta_m = 0.005$ radians. Tables 1 through 6, given in the Appendix, present the results for images without noise. Table 7 presents results when one of the frames is corrupted with uncorrelated noise of varying strength. Table 8 presents results when one of the frames is corrupted with correlated noise of varying strength.

The number of samples for the FOE/C during the entire search (combined global and local) was about 60, 160 and 375 (Tables 1, 2, and 3, respectively). There are approximately 200,000 operations/FOE/feature, bringing the total computation time to roughly 1 second per feature per position of the FOE/C (on the VAX 11-780). The global search for the minimum of the error surface is the time-consuming part of the search. For example, in experiments given in Table 3, they required almost 90% of the computation time). We would like to stress here, however, that the algorithm and the environment in which it was run were designed for programming flexibility in an experimental development process, and not in any way optimised for speed. Thus, the computational speed can be greatly improved.

3.2 Discussion of Experimental Results

Before we consider the results of individual experiments, several general remarks about the shape of the error surface for a camera moving in a static environment are in order:

1. The greater the number of the features, the "smoother" the error surface is expected to be.
2. The use of more features increases the computation time proportionally.
3. With fewer features, the percent of the contribution of one feature to the total error value is greater, and if some features are "weak", or if their number is small, the error surface is rougher.
4. The closer the feature is to the FOE/C, the less reliable is its contribution.
5. Features further from the FOE/C usually have larger displacements (although displacement is also a function of the depth of the environmental point) and therefore predict more accurately the orientation of the radial line on which the FOE/C should lie.
6. Since the position of the FOE/C is unknown, the features should be spaced more or less uniformly throughout the image plane.

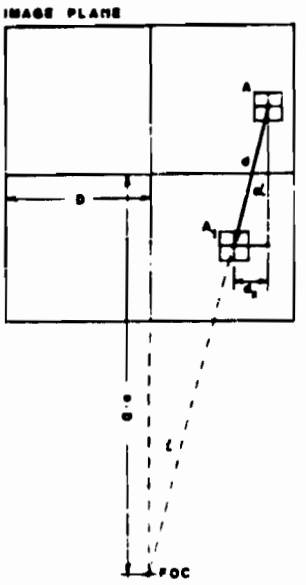
Let us consider for a moment the difficulties with the case of translational motion parallel (or almost parallel) to the image plane (i.e., the FOE/C is $\approx 90°$ to the direction of the line of sight), which requires a highly sensitive search in order to find the minimum of global error. In this case the FOE/C is far away from the center of the image plane (Figure 4).

Assume a window of size $w \cdot w$ is centered at (x, y) and moved a distance d towards the FOC (analysis is similar for the FOE). If the distance of the FOC is $n \cdot D$, $n \gg 1$, where $2D \cdot 2D$ is the size of the image plane, then according to Figure 4 we have:

$$d_o/d = x/(n \cdot D + y) \approx x/n \cdot D$$

or for $x \approx D$,

$$d_o \approx d/n.$$



When the FOC is n times the image size away from the image plane, a displacement of a feature d units towards the FOC causes the lateral displacement of the feature of roughly $d_o \approx d/n$.

Figure 4: Search for a distant FOC.

For example, if the FOE corresponds to the translation of 75° or 85° away from the line of sight, then $n \approx 4$ and $n \approx 10$, respectively. From these equations we can see that the correlation function must be able to detect a change of feature position in *the x-direction* of the order of d_x. However, d_x is expected to be very small when the position of the FOE/C is far from the image center, since the size of the displacement d is limited, at best, by the image size. More importantly, the displacement is usually assumed to be small to avoid ambiguities in the matching process. If the correlation function is not able to detect changes of size d_x in the displacement of a feature in the lateral direction, then the position of the FOE/C is unknown with an angular uncertainty of d_x/d. When the FOE/C is closer to the center of the image, the angular resolution of the method is also limited by the ability of the correlation function to detect such small lateral differences during feature displacements, but the relative error in the position of the FOE/C is much smaller.

The sensitivity of the correlation function also depends on the size of the window and the type of the correlation function. It is clear that the averaging nature of correlation function (being a sum of products) works against its sensitivity. Thus, windows that are larger do not necessarily imply better results.

The tables presented in the Appendix are considered next. In these tables, the second column represents the directions (in polar coordinates) in which the camera is actually translated and hence are the values which the algorithm is supposed to recover. We refer to these directions as "correct" values. The third, fourth and fifth columns specify deviations (in degrees) from the correct values from experimental runs for 4, 8 and 16 feature points, respectively.

In some cases the local search fails and is not possible to recover the correct axis of translation. These cases are marked with an A (for ambiguous) in the tables. Tables 1 through 3 show that the ambiguous results appear mostly for motions almost perpendicular to the line of sight, where small fluctuations in the error surface are most probably due to the decreasing sensitivity of the correlation function to small displacements. In these cases, the actual value returned by the algorithm was the coarse grid sampling point with the smallest error. Because of the noisy nature of the error surface, the finer grained local search cannot utilise the assumption of the convexity of the error surface in order to find the total minimum. Rather, the local search returns a value close to the value returned by the global search. Note that decreasing sparseness of the global search results in a smaller number of ambiguous values (compare Tables 1, 2, 3). Thus, failure to recover the correct axis is indeed due to the local part of the search, and therefore the results are not really a failure of the FOE/C method. (That is why we label them as ambiguous and not as errors.)

It is clear that 16 points seem to give adequately reliable results (see Table 3), with errors of only a few degrees from the correct axis. It is plausible that 32 or 64 feature points would improve the precision of the search for the FOE/C. In most tables results shown in one row suggest the improvement of accuracy as the number of features is increased. For four feature points the results are not accurate, but are relatively close to correct results, especially when camera translation is along the line of sight. In real-world images, this probably would not be a sufficient number of features.

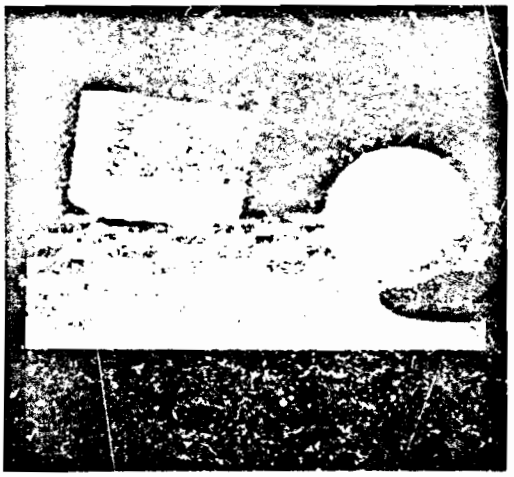
A finer division of the window displacements (sampling every third of a pixel; Table 4) does not produce much improvement with respect to the default set of parameters, demonstrating that the correlation measure between two windows has a relatively broad peak at the maximum.

Surprisingly, a window of smaller size (3x3) gave quite satisfactory results and in some cases recovered the axis of translation when larger windows failed to do so (compare Tables 1 and 5). Since the computation time with smaller windows is smaller, this window size might be used during the initial stages of computation. A window of size 11x11 did not show results any better than the default window size of 7 x 7.

Supplying an initial guess close to the correct axis of translation (Table 6) recovered the correct value in almost all cases. The search is done locally around the initial axis in a neighborhood δ_0 .The significance of this result is in showing that the correlation measure is able to detect the changes of axial positions to within a few degrees. However, one should not be misled by the "correctness" of the results. Since we have limited our search space here only to a very small δ_0 neighborhood around one of the points on the error surface, any returned value for the minimum will be not more than $\pm 2 \cdot \delta_0$ away from the initial guess. This estimate is a consequence of the fact that each time a new local minimum is found, the new area in which the search is continued has half the diameter of the old one ($\delta_0 + \delta_0/2 + \delta_0/4 + .. = 2 \cdot \delta_0$). Thus, the primary problem is to determine the correct neighborhood for local search and therefore the global search must have sufficient resolution.

Table 7 represents the results of the search for the correct translational axis when one of the images was corrupted with the uniform white noise. The corrupted image is shown in the Figure 5.

The program parameters were the same as those of Table 3, which was judged to represent the best results, in the set of experiments. The number of features was 16, therefore the last column of the Table 3 should be compared with the results in Tables 7 and 8. Noise was uniformly distributed, with an intensity range of 0 to 100, a mean of 50, and a standard deviation of 30. The gray level values of the original image ranged from a minimum of -88 to a



This image is obtained by adding white noise to the image shown in Figure 3.

Figure 5: Synthetic image used in experiments with added white noise.

maximum of 86, with an average of -54.0 and a standard deviation of 27.0. The intensity range of the noise added to this image was approximately 6%, 12%, 30% and 60% of the intensity range of the uncorrupted image. The results show the stability of the search in the presence of noise, with a graceful degradation of performance as the noise levels increase.

Finally, in Table 8 we present the results of runs on images with correlated noise. The correlated noise was obtained by averaging the white noise plane over a 5x5 neighborhood. This decreased the range of noise intensities to 24 to 72 (approximately a half of the white noise range) with a mean of 48.85 and a standard deviation of 5.8 (noticeably smaller than the standard deviation of the white noise; see Figure 6).

Tables 7 and 8 suggest that the algorithm is robust in the presence of noise. We are further exploring the relationship between noise levels, error values returned by the correlation function, and the overall performance of the algorithm.

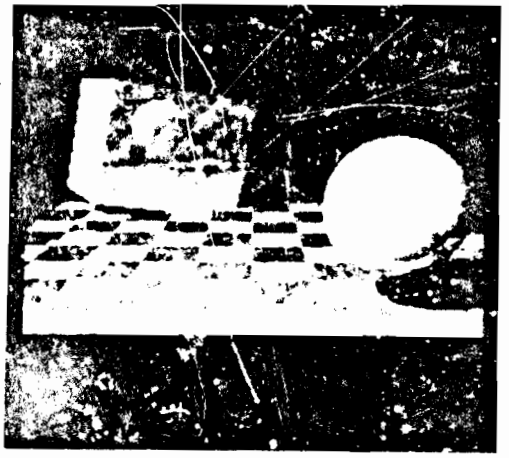
4. FURTHER WORK

This section describes some possible improvements to Lawton's algorithm and discusses search for the FOE/C from the point of view of a regularisation problem [POG84].

In many experiments done by Lawton the error surface is reported to exhibit an overall "smooth" behavior. This is to be expected for the following reasons. The correlation function between the two features is a relatively smooth function of the feature displacement (provided that the displacement is small, that there is no occlusion, and no dramatic local changes take effect). The total error function which is a sum of many smooth functions, should also be a smooth function. The assumption of slow temporal changes of direction of translation, often used in this kind of experiments, also implies a slow change of the position of the minimum of the total error on the image plane.

We have suggested that in some cases the error surface may not, in fact, be smooth. The hypothesis is that the local search may fail to return the correct location of the FOE/C because it has found a fake minima due to the presence of noise in the error surface. We are currently investigating this problem, and if the hypothesis proves to be correct, we intend to explore it using the notion of "regularisation". This idea is perhaps best expressed in the work by Poggio and Torre [POG84], although it is found in many early vision algorithms [GRI81, TER84]. In general, the theory assumes that many problems in early vision are of an "inverse", "ill-posed" type. Roughly speaking, ill-posed problems do not provide a unique solution and the space of acceptable solutions has to be restricted by choosing an appropriate "stabilisation" functional.

Our hypothesis, then, is that the problems that led to the failure to recover the correct translational axis are exactly due to the ill-posed nature of the problem and that a regularisation functional is needed to guide the search for the minimum of the error surface. The problem with the recovery of the correct translational axis can be partially



This image is obtained by adding correlated noise to the image shown in Figure 3.

Figure 6: Synthetic image used in experiments with added correlated noise.

overcome by imposing a smoothness constraint on the error surface. By forcing the error surface to be smooth small fluctuations in the error surface (due to the various problems discussed in this paper) should be eliminated and the search for the minimum facilitated by the smoother "valley" around the minimum. The information about the position of the minimum will be supported by the overall shape of the surface in this area, and therefore it will reflect the contribution of many points on the error surface.

The other advantage of this approach is that the fitting procedure can speed up the search process. Due to the smoothness assumption, one might start with a rather sparse set of the error values (FOE/C positions) and interpolate the values in the search for the global minimum. The search can then be localised and eventually repeated with a finer grid in a smaller region where the fitted surface has the minimum. Selection of the initial number of points in the sparse set of data requires compromise between the validity of the surface shape and the efficiency of the method.

5. SUMMARY

The report describes a set of experimental runs designed to determine the performance of a motion algorithm for detection of translational motion. The performance of the algorithm has been tested for various translational directions on a pair of synthetic images. The overall performance of the algorithm for cases where the camera motion is within 45° of the direction of the line of sight was very good: robust with respect to noise and accurate. The algorithm can return the translational axis to within a few degrees of the correct axis even if the images are subject to significant amounts of noise.

It was found that the algorithm is not adequately sensitive to translational almost normal to the line of sight. The problem was found to be not only one of insensitivity of the correlation function, but also one caused by the failure of the local search for the possible positions of the FOE/C in the case that the error surface around the minimum exhibits small fluctuations.

We have suggested an approach which would impose a constraint of the smoothness of the error surface. In a forthcoming paper we will demonstrate that this approach increases both speed and reliability of the search for the correct translational axis in cases where the tested algorithm had difficulties.

ACKNOWLEDGMENT

We would like to thank those people who contributed their time and effort to both the intelectual and system support environment within which this research was done, especially Daryl Lawton, Gilad Adiv, P. Anandan, Bob Heller, Michael Boldt, Brian Burns, and Seraj Bharwani.

APPENDIX

In this appendix a summary of runs is presented. The set of default values assumed for the parameters in the tested algorithm are as follows (for details see Section 2.3 and Section 3):

1. The step size (in pixels) of the correlation matching of each feature is one pixel.
2. The maximum distance a feature is moved in the search for the best correlation is 10 pixels.
3. The size of the window is 7x7 pixels.
4. The precision with which the translational axis was determined in the (α, β) space is $\delta_\infty = 0.005$ radians.
5. The density of the initial global search for the minimum of the error surface is roughly 45°. The number of steps in local search is determined by the size of δ_∞.

The tables summarise the results obtained from multiple runs of the algorithm under varying conditions. Tables 1 through 6 are images without noise, Table 7 presents results with uncorrelated uniform white noise, and Table 8 presents results with correlated noise. The following information is relevant for all the tables:

- The parameter that is changed from the default value, indicated above, is specified in the title of the table.
- The second column of the tables represents correct values for the direction of motion, where α is the angle between the projection of the translational axis on the Y-Z plane and the z-axis. β is the angle between the projection of the translational axis on the X-Z plane and the z-axis.
- Other columns give experimental results (in degrees) for deviation of angles α and β from the correct values.
- Symbol A stands for the "ambiguity" in the search for the right axis, a phenomenon discussed in Section 3.2.
- At the end of table a typical amount of CPU time on VAX 11-780 is given. This time represents time spent in both global and local search.

Table 1

Correct and experimental values for the translational axis, using the initial (default) set of parameters in the algorithm. (Global sampling approximately every 45°.)

Exp. No.	Correct axis (α, β)	Experiment 4 features	8 features	16 features
1	(0, 0)	(-0.2,-0.7)	(-0.7, 0.9)	(-0.3, 1.4)
2	(15, 0)	(13.6, 2.2)	(0.8, 1.2)	(1.2,-0.5)
3	(30, 0)	A	(3.4, 1.5)	(1.6, 0.5)
4	(45, 0)	A	A	A
5	(60, 0)	A	A	A
6	(75, 0)	A	A	A
7	(90, 0)	A	A	A
8	(120, 0)	(38.2, 0.0)	A	A
9	(150, 0)	(8.2, 0.0)	(8.2, 0.0)	(8.2, 0.0)
10	(180, 0)	A	A	A
11	(45, 90)	A	A	A
12	(0, 90)	A	A	A
13	(60, 30)	A	A	A
CPU	time	5 min.	10 min.	15 min.

Table 2

Increased density of points during global sampling over the unit sphere (approximately every 22.5 degrees, other parameters are the same as in Table 1).

Exp. No.	Correct axis (α, β)	Experiment 4 features	8 features	16 features
1	(0, 0)	(-0.2,-0.7)	(-0.7, 0.9)	(-0.3, 1.4)
2	(15, 0)	(13.6, 2.2)	(0.8, 1.2)	(1.2,-0.5)
3	(30, 0)	(20.7, 4.2)	(3.4, 1.5)	(1.7, 0.5)
4	(45, 0)	(16.7, 5.0)	(10.7, 3.4)	(-0.1, 0.3)
5	(60, 0)	A	A	A
6	(75, 0)	A	A	A
7	(90, 0)	A	A	A
8	(120, 0)	(21.3, 0.0)	(21.3, 0.0)	(21.3, 0.0)
9	(150, 0)	(18.7, 0.0)	(-8.7, 0.0)	(8.2, 0.0)
10	(180, 0)	A	A	A
11	(45, 90)	A	A	A
12	(0, 90)	A	A	A
13	(60, 30)	A	A	(-3.4,-30)
CPU	time	9 min.	18 min.	26 min.

Table 3

Increased density of points during global sampling over the unit sphere (approximately every 11.5 degrees, other parameters are the same as in Table 1).

Exp. No.	Correct axis (α, β)	Experiment 4 features	8 features	16 features
1	(0, 0)	(-0.7,-0.2)	(-0.7, 0.9)	(-0.3, 1.4)
2	(15, 0)	(13.7, 2.2)	(0.7, 1.4)	(1.4,-0.7)
3	(30, 0)	(20.7, 4.3)	(3.4, 1.5)	(1.4, 0.1)
4	(45, 0)	(16.7, 5.0)	(5.9, 0.2)	(-0.1, 0.3)
5	(60, 0)	A	(18.1,19.7)	(3.7, 0.0)
6	(75, 0)	A	A	A
7	(90, 0)	A	A	A
8	(120, 0)	(21.3, 0.0)	(9.8, 0.0)	(9.8, 0.0)
9	(150, 0)	(18.7, 0.0)	(-8.7, 0.0)	(8.2, 0.0)
10	(180, 0)	A	A	A
11	(45, 90)	A	A	A
12	(0, 90)	A	A	A
13	(60, 30)	A	A	(-3.4,-30)
CPU	time	30 min.	45 min.	55 min.

Table 4

Displacements between the windows are 1/3 of the pixel size. Other parameters are the same as in Table 1.

Exp. No.	Correct axis (α, β)	Experiment 4 features	8 features	16 features
1	(0, 0)	(-1.2, 0.0)	(-1.7, 0.0)	(-0.3, 2.1)
2	(15, 0)	(8.4, 1.9)	(1.5, 2.1)	(1.2,-0.3)
3	(30, 0)	A	(-0.4, 1.2)	(-3.3,-0.3)
4	(45, 0)	A	A	A
5	(60, 0)	A	A	A
6	(75, 0)	A	A	A
7	(90, 0)	A	A	A
8	(120, 0)	(38.2, 0.0)	A	A
9	(150, 0)	(8.2, 0.0)	(8.2, 0.0)	(8.2, 0.0)
10	(180, 0)	A	A	A
11	(45, 90)	A	A	A
12	(0, 90)	A	A	A
13	(60, 30)	A	A	A
CPU	time	11 min.	15 min.	23 min.

Table 5

Window size is 3x2. Other parameters are the same as in Table 1.

Exp. No.	Correct axis (α, β)	Experiment 4 features	8 features	16 features
1	(0, 0)	(-2.4, 0.0)	(-2.2, 0.5)	(-1.2, 1.9)
2	(15, 0)	(-1.8, 0.0)	(2.5, 2.2)	(-1.5,-2.1)
3	(30, 0)	(-6.4, 0.0)	(-6.1,-0.2)	(-4.3,-2.1)
4	(45, 0)	(1.7, 0.2)	A	(0.5, 1.0)
5	(60, 0)	(-3.9, 1.0)	(-8.5, 0.7)	A
6	(75, 0)	A	A	A
7	(90, 0)	(43.6,-0.2)	A	A
8	(120, 0)	(13.6,-0.2)	A	A
9	(150, 0)	(8.2, 0.0)	(-11.7, 0.0)	(8.2, 0.0)
10	(180, 0)	A	A	A
11	(45, 90)	A	A	A
12	(0, 90)	A	A	A
13	(60, 30)	(-40,-15.5)	A	A
CPU	time	6.5 min.	6.8 min.	7.3 min.

Table 6

Initial guess for the translation axis is supplied as an input parameter to the algorithm. Other parameters are the same as in Table 1.

Exp. No.	Correct axis (α, β)	Experiment 4 features	8 features	16 features
1	(0, 0)	(-0.7,-0.2)	(-0.5, 0.9)	(-0.4, 1.4)
2	(15, 0)	(13.7, 2.2)	(0.8, 1.3)	(1.3,-0.5)
3	(30, 0)	(21.0, 4.3)	(3.5, 1.6)	(-2.8,-0.4)
4	(45, 0)	(16.1, 4.9)	(10.7, 3.4)	(0.2, 0.4)
5	(60, 0)	(15.9,10.5)	(8.7, 7.6)	(3.9, 0.0)
6	(75, 0)	F*	(25.2,-8.3)	(2.8,-4.0)
7	(90, 0)	(13.2, 1.8)	F	F
8	(120, 0)	(20.7,-0.2)	(8.0,-3.4)	(7.5,-1.6)
9	(150, 0)	(0.1, 0.0)	(0.1, 0.0)	(0.1, 0.0)
10	(180, 0)	A	A	A
11	(45, 90)	A	A	A
12	(0, 90)	A	A	A
13	(60,30)	(11.8,28.1)	(-7.1, 2.3)	(9.9,27.3)
CPU	time	11 min.	13.5 min.	18 min.

* (fails to converge)

Table 7

Addition of uniform, uncorrelated, white noise. Otherwise, same parameters as in the experiments in Table 3, with 16 features.

Exp No.	Correct (α, β)	Exper. 6%*	12%	30%	60%
1	(0, 0)	(0.2, 1.4)	(-0.7, 1.7)	(1.2, 1.9)	(0.4,11.5)
2	(15, 0)	(1.8,-0.2)	(2.1,-0.3)	(-0.7,1.0)	(0.9,-2.1)
3	(30, 0)	(-2.5,-0.9)	(-3.6,-0.7)	(-2.8,-0.9)	(-3.5,-2.1)
4	(45, 0)	(1.2, 0.9)	(0.5, 0.3)	(-6.2,-2.2)	(2.7, 2.6)
5	(60, 0)	(6.5, 1.7)	(4.6, 0.7)	(4.6, 0.0)	(9.8,5.6)
6	(75, 0)	A	A	A	A
7	(90, 0)	A	A	A	A
8	(120, 0)	(9.8, 0.0)	(9.8, 0.0)	(9.8, 0.0)	(9.8, 0.0)
9	(150, 0)	(8.2, 0.0)	(-1.0, 0.0)	(-1.0, 0.0)	(14.6,-5.2)
10	(180, 0)	A	A	A	A
13	(60, 30)	(3.4,-30)	(3.4,-30)	(-3.7,-30)	(3.4,-30)
CPU	time	1 hour	1 hour	1 hour	1 hour

* (of uncorrelated noise)

Table 8

Addition of uniform, correlated, noise. Otherwise, same parameters as in experiments in the Table 3, with 16 features.

Exp No.	Correct (α, β)	Exper. 6%*	12%	30%	60%
1	(0, 0)	(-1.4, 1.0)	(0.7, 0.0)	(0.0, 1.9)	(-0.5, 3.6)
2	(15, 0)	(1.6,-0.5)	(-1.9,0.5)	(-1.3,-2.9)	(-1.4,-1.0)
3	(30, 0)	(-3.9,-0.7)	(-3.6,-0.7)	(-4.9, 3.0)	(-5.1,-0.9)
4	(45, 0)	(-0.1, 0.3)	(-5.2,-1.4)	(-1.9, 0.3)	(3.9, 5.4)
5	(60, 0)	(3.7, 0.3)	(-0.5,-1.4)	(5.7, 2.4)	(2.9, 1.0)
6	(75, 0)	A	A	A	A
7	(90, 0)	A	A	A	A
8	(120, 0)	(9.8, 0.0)	(9.8, 0.0)	(3.7, 0.0)	(9.8, 0.0)
9	(150, 0)	(8.2, 0.0)	(-1.0, 0.0)	(8.2, 0.0)	(8.2, 0.0)
10	(180, 0)	A	A	A	A
13	(60, 30)	(3.4,-30)	(-3.7,-30)	(3.4,-30)	(3.4,-3.0)
CPU	time	1 hour	1 hour	1 hour	1 hour

* (of correlated noise)

REFERENCES

[GIB50]
J. J. Gibson, *The Perception of the Visual World*, Cambridge, Mass., Riverside, 1950.

[GRI81]
W. E. L. Grimson, *A Computational Theory of Visual Surface Interpolation*, MIT, A.I. Memo 613, 1981.

[HAN84]
Allen R. Hanson and Edward M. Riseman, *A Summary of Image Understanding Research at the University of Massachusetts*, Technical Report 83-35, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts 01003, October 1983.

[LAW83]
Daryl T. Lawton, *Processing Translational Motion Sequences*, Computer Graphics and Image Processing, Vol. 22, pp. 116-144, 1983.

[LAW84]
Daryl T. Lawton, *Processing Dynamic Image Sequences from a Moving Sensor* Ph. D. Dissertation. University of Massachusetts, Amherst, MA 01003. Also, Technical Report 84-05, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts 01003, February 1984.

[MOR77]
H. P. Moravec, *Towards Automatic Visual Obstacle Avoidance*, Proceedings of the 5th IJCAI, MIT, Cambridge, MA, 1977, p. 584.

[POG84]
T. Poggio and V. Torre, *Ill-Posed Problems and Regularization Analysis in Early Vision* A.I. Memo 773, MIT, Cambridge, MA, 1984.

[TER84]
D. Tersopoulos, *Computation of Visible-Surface Representations*, Ph. D. Dissertation, Massachusetts Institute of Technology, Jan. 1983.

[WHI80]
T. Whitted, *An Improved Illumination Model for Shaded Display*, Communications of the ACM, 23(6), June 1980, pp. 343-349.

[WIL80]
T. D. Williams, *Depth from Camera Motion in a Real World Scene*, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-2(6), November 1980, pp. 511-516.

GC970145-43

INHERENT AMBIGUITIES IN RECOVERING 3-D MOTION AND STRUCTURE FROM A NOISY FLOW FIELD

Gilad Adiv

Computer and Information Science Department
University of Massachusetts
Amherst, MA 01003

Abstract

One of the major areas in research on dynamic scene analysis is recovering 3-D motion and structure from optical flow information. Two problems which may arise due to the presence of noise in the flow field are presented in this paper. First, motion parameters of the sensor or a rigidly moving object may be extremely difficult to estimate because there may exist a large set of significantly incorrect solutions which induce flow fields similar to the correct one. The second problem is in the decomposition of the environment into independently moving objects. Two such objects may induce optical flows which are compatible with the same motion parameters and, hence, there is no way to refute the hypothesis that these flows are generated by one rigid object. These ambiguities are inherent in the sense that they are algorithm-independent. Using a mathematical analysis, we characterise situations where these problems are likely to arise. A few examples demonstrate the conclusions. Constraints and parameters which can be recovered even in ambiguous situations are presented.

1. Introduction

Dynamic visual information can be produced by a sensor moving through the environment and/or by independently moving objects in the visual field. The interpretation of such information consists of detecting moving objects, recovering the motion parameters of the sensor and each moving object, and structure determination. The results of this interpretation can be used to control behaviour, as in robotics or navigation. They can also be integrated, as an additional knowledge source, into an image understanding system, such as the VISIONS system [HAN78].

The most common approach for the analysis of visual motion is based on two phases: computation of an optical flow field and interpretation of this field. In the present discussion, the term 'optical flow field' refers to a 'velocity field', composed of vectors describing the instantaneous velocity of image elements. The second phase, i.e., the interpretation of the optical flow field, is the main concern of this paper. The information in only one flow field, as opposed to a time sequence of such fields, is assumed to be given.

Flow fields generated by existing techniques are noisy and partially incorrect, especially near occlusion or motion boundaries (see the discussion in [ULL81]). Two problems may arise due to the presence of noise in the flow field. First, motion parameters of the sensor or a moving object may be extremely difficult to estimate because there may exist a large set of significantly incorrect solutions which induce flow fields similar to the correct one. The second problem, which is closely related to the first one, is in the decomposition of the environment into independently moving objects. Two such objects may induce optical flows which are compatible with the same motion parameters and, hence, there is no way to refute the hypothesis that these flows are generated by one rigid object. These ambiguities are inherent in the sense that they are algorithm-independent.

In this paper we will employ mathematical analysis to characterise situations where these problems are likely to arise. A few examples will demonstrate the conclusions. Constraints and parameters which can be recovered even in ambiguous situations, as well as appropriate modifications of the interpretation goals, will be presented.

2. Equations Relating the Optical Flow to 3-D Motion and Structure

2.1 The General Case

Let (X, Y, Z) represent a cartesian coordinate system which is fixed with respect to the camera (see Figure 2.1), and let (x, y) represent a corresponding coordinate system of a planar image. The image is assumed to be a square and the field of view (FOV) is defined to be the visual angle corresponding to each side of the image, which, therefore, is $2\tan(fov/4)$ focal units. The focal length, from the nodal point C to the image, is assumed to be known and, without loss of generality, it can be normalised to 1. Thus, the perspective projection (x, y) on the image of a point (X, Y, Z) in the environment is:

$$x = X/Z, \qquad y = Y/Z. \qquad (2.1a,b)$$

The motion, relative to the camera, of a rigid object in the scene can be decomposed into two components: trans-

This work was supported by DARPA under Grant N00014-82-K-0464. The author is now with Rafael, POB 2250, Haifa 31021, Israel.

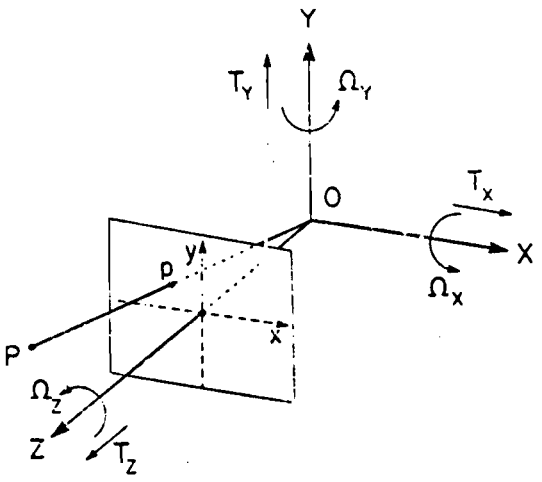


Figure 2.1 (redrawn from [LON80]): A coordinate system (X,Y,Z) attached to the camera, and the corresponding image coordinates (x,y). The image position p is the perspective projection of the point P in the environment. $\underline{T} = (T_X, T_Y, T_Z)$ and $\underline{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)$ represent the relative translation and rotation of a given object in the scene.

lation $\underline{T} = (T_X, T_Y, T_Z)$ and rotation $\underline{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)$ (these symbols represent instantaneous spatial velocities). If (X,Y,Z) are the instantaneous camera coordinates of a point on the object, then the corresponding projection (x,y) on the image moves with a velocity (α,β), where [LON80]:

$$\alpha = -\Omega_X xy + \Omega_Y(1+x^2) - \Omega_Z y + (T_X - T_Z x)/Z, \quad (2.2a)$$

$$\beta = -\Omega_X(1+y^2) + \Omega_Y xy + \Omega_Z x + (T_Y - T_Z y)/Z. \quad (2.2b)$$

Notice that (α,β) can be represented as the sum

$$(\alpha,\beta) = (\alpha_R,\beta_R) + (\alpha_T,\beta_T), \quad (2.3)$$

where (α_R,β_R) and (α_T,β_T) are, respectively, the rotational and translational components of the velocity field:

$$\alpha_R = -\Omega_X xy + \Omega_Y(1+x^2) - \Omega_Z y, \qquad \alpha_T = (T_X - T_Z x)/Z, \quad (2.4a,b)$$

$$\beta_R = -\Omega_X(1+y^2) + \Omega_Y xy + \Omega_Z x, \qquad \beta_T = (T_Y - T_Z y)/Z. \quad (2.4c,d)$$

Given a flow field, we wish to estimate the recoverable motion parameters of each rigid object, relative to the camera. These parameters are the rotation parameters $(\Omega_X,\Omega_Y,\Omega_Z)$, and the direction of the translation vector defined by the unit vector $\underline{U} = \underline{T}/r$ where r is the length of $\underline{T}$. Notice that a stationary environment is considered to be a rigid object moving relative to the camera. Once the motion parameters are recovered, it is also possible to estimate the relative depth, $Z(x,y)/r$, corresponding to each pixel (x,y) where a flow vector is defined, unless $r = 0$ or the location of the vector is exactly in the focus of expansion (FOE).

2.2 The Planar Case

In this section we examine the flow field induced by a rigid motion of a planar surface. Excluding the degenerate case in which the same plane contains both the surface and the nodal point (and, therefore, the corresponding region in the image is a straight line), the surface can be represented by the equation

$$k_X X + k_Y Y + k_Z Z = 1. \quad (2.5)$$

The coefficients k_X, k_Y and k_Z can be any real numbers, except the case in which all of them are zero. Introducing the notations $\underline{k} = (k_X, k_Y, k_Z)$ and $\underline{l} = r\underline{k}$, and using equation (2.1), we obtain:

$$r/Z = l_X x + l_Y y + l_Z. \quad (2.6)$$

Using the identity $\underline{T} = r\underline{U}$ and substituting (2.6) in (2.2), we realise that, given a relative motion $\{\underline{T},\underline{\Omega}\}$, the flow field is:

$$\alpha = a_1 + a_2 x + a_3 y + a_7 x^2 + a_8 xy, \quad (2.7a)$$

$$\beta = a_4 + a_5 x + a_6 y + a_7 xy + a_8 y^2, \quad (2.7b)$$

where:

$$a_1 = \Omega_Y + l_Z U_X, \quad (2.8a)$$
$$a_2 = l_X U_X - l_Z U_Z, \quad (2.8b)$$
$$a_3 = -\Omega_Z + l_Y U_X, \quad (2.8c)$$
$$a_4 = -\Omega_X + l_Z U_Y, \quad (2.8d)$$
$$a_5 = \Omega_Z + l_X U_Y, \quad (2.8e)$$
$$a_6 = l_Y U_Y - l_Z U_Z, \quad (2.8f)$$
$$a_7 = \Omega_Y - l_X U_Z, \quad (2.8g)$$
$$a_8 = -\Omega_X - l_Y U_Z. \quad (2.8h)$$

Equations (2.7) represent what we shall call a ☤ *transformation*. They describe a 2-D motion in the image plane, represented by the 8 parameters $a_1,\ldots,a_8$. Note that a similar representation of the optical flow produced by a moving planar surface is introduced in [WAX83].

3. Ambiguity in Determining Motion Parameters of a Rigid Object

3.1 Introductory Discussion

Given a flow field induced by a rigid object, it will be shown that in certain situations the flow field induced by totally incorrect motion and structure may be similar to the correct one. In the presence of noise which is statistically larger than the difference between these flow fields, it may be impossible to obtain reasonably accurate estimates of

the motion parameters. The influence of certain factors on this ambiguity will be analysed.

Let us start by examining the cases of pure rotation and pure translation. In a purely rotational motion the flow field is represented by equations (2.4a,c) which can be rewritten as

$$\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \begin{pmatrix}-xy\\ -1-y^2\end{pmatrix}\Omega_X + \begin{pmatrix}1+x^2\\ xy\end{pmatrix}\Omega_Y + \begin{pmatrix}-y\\ x\end{pmatrix}\Omega_Z. \tag{3.1}$$

Thus, each rotation parameter has a distinct signature in the flow field, and in most cases it can reliably be recovered.

In a purely translational motion, the direction of translation is represented by the focus of expansion (FOE) which, in this case, is the intersection of the straight lines corresponding to the flow vectors. Usually, the FOE can be robustly recovered [LAW84], unless the absolute value of the translation is small relative to the distance of the surface from the observer, in which case the flow vectors are small and the determination of the corresponding intersection is sensitive to noise.

In the general case, an ambiguity in determining the motion parameters becomes a much more severe problem because rotation and translation may induce similar flows. To demonstrate this, let us examine the case of a planar surface which is parallel to the image plane, and denote by d the distance of this plane from the camera. We wish now to compare the flow field generated by a purely translational motion $(P_X, P_Y, 0)$ to the flow field generated by the purely rotational motion $(-P_Y/d, P_X/d, 0)$. The flow field in the first case is

$$\begin{pmatrix}\alpha_T\\ \beta_T\end{pmatrix} = 1/d\begin{pmatrix}P_X\\ P_Y\end{pmatrix}, \tag{3.2}$$

while in the second case the flow is given by

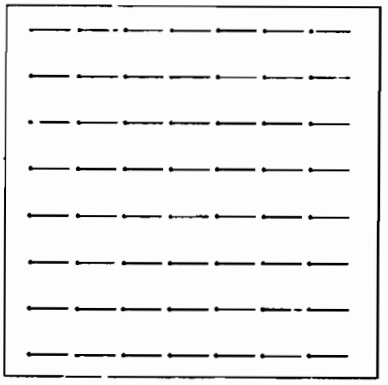
$$\begin{pmatrix}\alpha_R\\ \beta_R\end{pmatrix} = 1/d\begin{pmatrix}xyP_Y + (1+x^2)P_X\\ (1+y^2)P_Y + xyP_X\end{pmatrix}. \tag{3.3}$$

Hence,

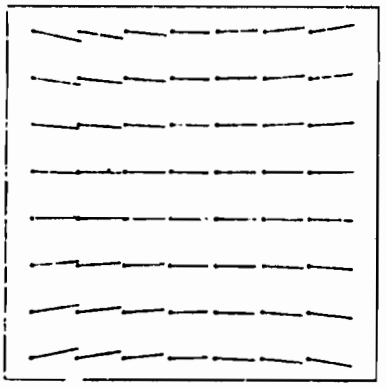
$$\begin{pmatrix}\alpha_R\\ \beta_R\end{pmatrix} - \begin{pmatrix}\alpha_T\\ \beta_T\end{pmatrix} = 1/d\begin{pmatrix}xyP_Y + x^2P_X\\ y^2P_Y + xyP_X\end{pmatrix}. \tag{3.4}$$

If the field of view (FOV) is small, then the second-order terms of the image coordinates, x and y, are small, and the difference between the flow fields is small as well (see Figure 3.1). In such a case it may be very difficult, in the presence of noise, to distinguish between these fields and to determine whether the motion is purely translational, purely rotational or a combination of both. Note, however, that if the FOV is large, then the second-order components of the flow field are relatively large and, therefore, the ambiguity is more likely to be resolved.

Ambiguity in determining the motion parameters is affected not only by the FOV, but also by variations in the surface structure. This can be concluded from the work of



(a)



(b)

Figure 3.1: The flow field in (a) is purely translational, whereas the flow field in (b) is purely rotational. In both cases the field of view is 60°. Note the similarity of the flow fields in the central portion of the image, where the values of x and y are small. On the other hand, the difference between the flow fields is not negligible near the boundary of the image.

Rieger and Lawton [RIE83], who examined the case of large discontinuities in the depth map. In this case the differences between flow vectors near occlusion boundaries are oriented towards the FOE of the translational field and, therefore, the ambiguity in distinguishing between the translational and rotational components can be resolved.

In addition, it has been experimentally shown [PRA80, LON81, FAN83b] that the accuracy of the estimated 3-D motion parameters is improved when the translational component of the motion is large relative to the distance of the object from the sensor. The results can also be improved by using a large number of flow vectors [ROA80, TSA84], and by increasing the size of the region containing these vectors [PRA80, FAN83a].

To summarise, ambiguity (or, using another term, instability) in determining 3-D information can be expected if the FOV, the depth variations, and the ratio of the translation to the distance of the object from the camera are all small. In addition, local techniques, in which only the information in a small region of the image is utilised, are more sensitive to noise. In the next section we will employ a mathematical analysis in order to show that these conditions, as well as other conditions, generally contribute to ambiguity in recovering 3-D motion and structure. Combined together these are sufficient conditions for such instability. Note that the error analyses existing in the literature are experimental and algorithm-dependent, whereas here we develop a mathematical and algorithm-independent analysis.

3.2 Mathematical Analysis

3.2.1 A planar surface

In this section we restrict ourselves to flow fields induced by a rigid motion of a planar surface given by equation (2.5): $k_X X + k_Y Y + k_Z Z = 1$. Since

$$Z = \frac{1}{k_Z} - \frac{k_X}{k_Z} X - \frac{k_Y}{k_Z} Y, \tag{3.5}$$

$1/k_Z$, denoted by d, is the distance from the camera to the surface along the line of sight, and the values $-k_X/k_Z$ and $-k_Y/k_Z$, denoted, respectively, by s_X and s_Y, represent the slopes of the surface relative to the image plane.

In the following analysis, the image is assumed to be square and the FOV is defined to be the visual angle corresponding to each side of the image, which, therefore, is $2\tan(fov/2)$ focal units. The region $\mathcal{R}$ corresponding to the perspective projection of the planar patch on the image is contained in a square for which the proportion between its side and the image side is γ, where $0 < \gamma \leq 1$. For simplicity, the center of this square is assumed to be $(0,0)$, but even if this is not the case, results similar to those which we will obtain in this section can be derived by expanding the flow equations around this center. γ, which we shall call the *locality factor*, will be small if the ratio of the object size to its distance from the camera is small relative to the image size (in focal units), or if a technique based on a local analysis of the flow field is employed.

The flow field generated by the motion parameters $\{\underline{U}, \underline{\Omega}\}$ can be described by a Ψ transformation (equations (2.7)) with the 8 coefficients given in equations (2.8). Employing the constraint $U_X^2 + U_Y^2 + U_Z^2 = 1$ in addition, we obtain 9 non-linear equations with 9 unknowns: $\underline{U}$, $\underline{\Omega}$ and $\underline{l}$. Usually, these equations have 2 sets of solutions [TSA84, WAX83], where, of course, only one of them is the correct one. Let us now denote by $\hat{\underline{U}}$, $\hat{\underline{\Omega}}$ and $\hat{\underline{l}}$ estimates of the motion and structure parameter values. We will show that in some situations, vectors $\hat{\underline{U}}$ significantly different from the corresponding values in each of the two exact solutions produce flow fields which are very similar to the correct one, if combined with appropriate values of $\hat{\underline{\Omega}}$ and $\hat{\underline{l}}$.

The basic idea is that if the region $\mathcal{R}$ is rather small (in focal units) and l_X, l_Y are not large, then, based on equations (2.7) and (2.8g,h), a change in $\underline{U}$ has only a small effect on the second-order components of the flow field. Therefore, given an arbitrary $\hat{\underline{U}}$, we concentrate on the lower-order components, and try to find $\hat{\underline{\Omega}}$ and $\hat{\underline{l}}$ such that the correct values of the coefficients $a_1, \ldots, a_6$ will be maintained. This will lead to hypothesised values of the motion and structure parameters. We can substitute these parameters in the expressions for a_7 and a_8 and measure the deviation of the obtained values from the correct ones. These deviations determine what we shall call the *error field*, that is, the discrepancy between the correct flow field and that predicted from the hypothesised parameters. Note that this error field is actually an upper bound to the 'minimal' error field which could be obtained for the same $\hat{\underline{U}}$ by optimising the values of $\hat{\underline{\Omega}}$ and $\hat{\underline{l}}$ across all the eight coefficients instead of $a_1, \ldots, a_6$.

Given a vector $\hat{\underline{U}}$, the equations (2.8a) to (2.8f) associated with the coefficients $a_1, \ldots, a_6$ produce six linear equations with six unknowns: $\hat{\Omega}_X$, $\hat{\Omega}_Y$, $\hat{\Omega}_Z$, $\hat{l}_X$, $\hat{l}_Y$ and $\hat{l}_Z$. These equations can be represented by

$$F\underline{u} = \underline{a}, \tag{3.6a}$$

where

$$F = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & \hat{U}_Y \\ 0 & 1 & 0 & 0 & 0 & \hat{U}_X \\ 0 & 0 & -1 & 0 & \hat{U}_X & 0 \\ 0 & 0 & 1 & \hat{U}_Y & 0 & 0 \\ 0 & 0 & 0 & \hat{U}_X & 0 & -\hat{U}_Z \\ 0 & 0 & 0 & 0 & \hat{U}_Y & -\hat{U}_Z \end{pmatrix}, \tag{3.6b}$$

$$\underline{u} = \begin{pmatrix} \hat{\Omega}_X \\ \hat{\Omega}_Y \\ \hat{\Omega}_Z \\ \hat{l}_X \\ \hat{l}_Y \\ \hat{l}_Z \end{pmatrix} \quad \text{and} \quad \underline{a} = \begin{pmatrix} a_4 \\ a_1 \\ a_3 \\ a_5 \\ a_2 \\ a_6 \end{pmatrix}. \tag{3.6c,d}$$

A unique solution is guaranteed if the determinant of F, denoted by D, is non-zero. As can easily be verified

$$D = \hat{U}_Z(\hat{U}_X^2 + \hat{U}_Y^2). \tag{3.7}$$

Thus, if the translation vector is not exactly perpendicular ($\hat{U}_X = \hat{U}_Y = 0$), or parallel ($\hat{U}_Z = 0$), to the image plane, then there exist $\hat{\Omega}$ and $\hat{l}$ which keep the correct values of $a_1, \ldots, a_8$. Note, however, that this solution may still be not physically realisable if the depth constraint $Z > 0$ is not satisfied by one of the depth values predicted by the vector $\hat{l}$. We will return to this problem in the end of this section.

The solution of the equations set (3.6) is, if $D \neq 0$,

$$\underline{a} = F^{-1}\underline{a}. \tag{3.8}$$

To find F^{-1} we employ the decomposition technique in [RAL65]. We represent F by

$$F = \begin{pmatrix} F_1 & F_2 \\ F_3 & F_4 \end{pmatrix}, \tag{3.9}$$

where each of F_i $(i = 1,2,3,4)$ is a 3×3 matrix. Then

$$F^{-1} = \begin{pmatrix} G_1 & G_2 \\ G_3 & G_4 \end{pmatrix}, \tag{3.10}$$

where

$$G_1 = F_1^{-1} - F_1^{-1}F_2G_3, \tag{3.11a}$$

$$G_2 = -F_1^{-1}F_2G_4, \tag{3.11b}$$

$$G_3 = -G_4F_3F_1^{-1} \tag{3.11c}$$

and

$$G_4 = (F_4 - F_3F_1^{-1}F_2)^{-1}. \tag{3.11d}$$

Necessary and sufficient conditions for applying this technique is that F_1 and $F_4 - F_3F_1^{-1}F_2$ are non-singular matrices. These conditions are satisfied if F is non-singular.

Employing the decomposition technique, we can now obtain

$$F^{-1} = \frac{1}{D}\begin{pmatrix} -D & 0 & \hat{U}_X\hat{U}_Y^2 & \hat{U}_X\hat{U}_Y^2 & -\hat{U}_Y^2 & -\hat{U}_X^2\hat{U}_Y \\ 0 & D & -\hat{U}_X^2\hat{U}_Y & -\hat{U}_X^2\hat{U}_Y & \hat{U}_X\hat{U}_Y^2 & \hat{U}_X^3 \\ 0 & 0 & -\hat{U}_X^2\hat{U}_Z & \hat{U}_X^2\hat{U}_Z & -\hat{U}_X\hat{U}_Y\hat{U}_Z & \hat{U}_X\hat{U}_Y\hat{U}_Z \\ 0 & 0 & \hat{U}_Y\hat{U}_Z & \hat{U}_Y\hat{U}_Z & \hat{U}_X\hat{U}_Z & -\hat{U}_X\hat{U}_Z \\ 0 & 0 & \hat{U}_X\hat{U}_Z & \hat{U}_X\hat{U}_Z & -\hat{U}_Y\hat{U}_Z & \hat{U}_Y\hat{U}_Z \\ 0 & 0 & \hat{U}_X\hat{U}_Y & \hat{U}_X\hat{U}_Y & -\hat{U}_Y^2 & -\hat{U}_X^2 \end{pmatrix} \tag{3.12}$$

Substituting the definitions of $\underline{a}$ and $\underline{\alpha}$ in (2.8),

$$\underline{a} = \begin{pmatrix} \hat{\Omega}_X \\ \hat{\Omega}_Y \\ \hat{\Omega}_Z \\ \hat{l}_X \\ \hat{l}_Y \\ \hat{l}_Z \end{pmatrix} = F^{-1}\begin{pmatrix} a_4 \\ a_1 \\ a_3 \\ a_5 \\ a_2 \\ a_6 \end{pmatrix} = F^{-1}\begin{pmatrix} -\Omega_X + U_Yl_Z \\ \Omega_Y + U_Xl_Z \\ -\Omega_Z + U_Xl_Y \\ \Omega_Z + U_Yl_X \\ U_Xl_X - U_Zl_Z \\ U_Yl_Y - U_Zl_Z \end{pmatrix}, \tag{3.13}$$

where $\underline{U}$, $\underline{\Omega}$ and $\underline{l}$ are the correct values of the motion and structure parameters. Multiplying F^{-1} by $\underline{\alpha}$, and using the notations $\epsilon_{XY} = \hat{U}_XU_Y - \hat{U}_YU_X$, $\epsilon_{XZ} = \hat{U}_XU_Z - \hat{U}_ZU_X$ and $\epsilon_{YZ} = \hat{U}_YU_Z - \hat{U}_ZU_Y$, we can derive:

$$\hat{\Omega}_X = \Omega_X + \frac{\hat{U}_Y\epsilon_{XY}(\hat{U}_Yl_X - \hat{U}_Xl_Y)}{\hat{U}_Z(\hat{U}_X^2 + \hat{U}_Y^2)} + \frac{\epsilon_{YZ}}{\hat{U}_Z}l_Z, \tag{3.14a}$$

$$\hat{\Omega}_Y = \Omega_Y + \frac{\hat{U}_X\epsilon_{XY}(\hat{U}_Xl_Y - \hat{U}_Yl_X)}{\hat{U}_Z(\hat{U}_X^2 + \hat{U}_Y^2)} - \frac{\epsilon_{XZ}}{\hat{U}_Z}l_Z, \tag{3.14b}$$

$$\hat{\Omega}_Z = \Omega_Z + \frac{\epsilon_{XY}(\hat{U}_Xl_X + \hat{U}_Yl_Y)}{\hat{U}_X^2 + \hat{U}_Y^2}, \tag{3.14c}$$

$$\hat{l}_X = \frac{(\hat{U}_XU_X + \hat{U}_YU_Y)l_X - \epsilon_{XY}l_Y}{\hat{U}_X^2 + \hat{U}_Y^2}, \tag{3.14d}$$

$$\hat{l}_Y = \frac{(\hat{U}_XU_X + \hat{U}_YU_Y)l_Y + \epsilon_{XY}l_X}{\hat{U}_X^2 + \hat{U}_Y^2} \tag{3.14e}$$

and

$$\hat{l}_Z = \frac{U_Z}{\hat{U}_Z}l_Z + \frac{\epsilon_{XY}(\hat{U}_Yl_X - \hat{U}_Xl_Y)}{\hat{U}_Z(\hat{U}_X^2 + \hat{U}_Y^2)}. \tag{3.14f}$$

The error field corresponding to the values $\hat{\underline{U}}$, $\hat{\Omega}$ and $\hat{l}$ is the deviation between the flow field predicted by these values and the correct field. Referring to equations (2.7), its value at the (x, y) pixel is:

$$\begin{pmatrix} \Delta\alpha_2 \\ \Delta\beta_2 \end{pmatrix} = \begin{pmatrix} \Delta a_7x^2 + \Delta a_8xy \\ \Delta a_7xy + \Delta a_8y^2 \end{pmatrix}, \tag{3.15}$$

where Δa_7 and Δa_8 are the errors induced in the coefficients a_7 and a_8. Recall now that the distance, denoted by d, from the camera to the surface along the line of sight is $1/k_Z$, and the slopes, denoted by s_X and s_Y, of the surface relative to the image plane are, respectively, $-k_X/k_Z$ and $-k_Y/k_Z$. Hence,

$$r/d = l_Z, \quad s_X = -l_X/l_Z \quad \text{and} \quad s_Y = -l_Y/l_Z. \tag{3.16a,b,c}$$

Thus, using equations (2.8g,h) and (3.14), we can obtain:

$$\Delta a_7 = (\hat{\Omega}_Y - \hat{U}_Z \hat{l}_X) - (\Omega_Y - U_Z l_X) \tag{3.17a}$$

$$= \frac{\hat{U}_X \epsilon_{XZ} + \hat{U}_Y \epsilon_{YZ}}{\hat{U}_X^2 + \hat{U}_Y^2} l_X + \frac{-\hat{U}_X \hat{U}_Y l_X + (1 - \hat{U}_Y^2) l_Y}{\hat{U}_Z (\hat{U}_X^2 + \hat{U}_Y^2)} \epsilon_{XY} - \frac{\epsilon_{XZ} l_Z}{\hat{U}_Z}$$

$$= \frac{-r}{d} \left[\frac{\hat{U}_X \epsilon_{XZ} + \hat{U}_Y \epsilon_{YZ}}{\hat{U}_X^2 + \hat{U}_Y^2} s_X + \frac{-\hat{U}_X \hat{U}_Y s_X + (1 - \hat{U}_Y^2) s_Y}{\hat{U}_Z (\hat{U}_X^2 + \hat{U}_Y^2)} \epsilon_{XY} + \frac{\epsilon_{XZ}}{\hat{U}_Z} \right]$$

and

$$\Delta a_8 = (-\hat{\Omega}_X - \hat{U}_Z \hat{l}_Y) - (-\Omega_X - U_Z l_Y) \tag{3.17b}$$

$$= \frac{\hat{U}_X \epsilon_{XZ} + \hat{U}_Y \epsilon_{YZ}}{\hat{U}_X^2 + \hat{U}_Y^2} l_Y + \frac{(\hat{U}_X^2 - 1) l_X + \hat{U}_X \hat{U}_Y l_Y}{\hat{U}_Z (\hat{U}_X^2 + \hat{U}_Y^2)} \epsilon_{XY} - \frac{\epsilon_{YZ} l_Z}{\hat{U}_Z}$$

$$= \frac{-r}{d} \left[\frac{\hat{U}_X \epsilon_{XZ} + \hat{U}_Y \epsilon_{YZ}}{\hat{U}_X^2 + \hat{U}_Y^2} s_Y + \frac{(\hat{U}_X^2 - 1) s_X + \hat{U}_X \hat{U}_Y s_Y}{\hat{U}_Z (\hat{U}_X^2 + \hat{U}_Y^2)} \epsilon_{XY} + \frac{\epsilon_{YZ}}{\hat{U}_Z} \right].$$

Therefore, if the translation is not large relative to the distance of the surface from the camera (i.e., r/d is small), and the surface is not very slanted (i.e., s_X and s_Y are small), then Δa_7 and Δa_8 are not large for vectors $\hat{\underline{U}}$ in a relatively large neighborhood of $\underline{U}$. If, in addition, the region $\mathcal{R}$ is small, then x and y are small as well, and thus the deviation $(\Delta\alpha_\theta, \Delta\beta_\theta)$ is very small. Under these conditions, any error surface corresponding to possible values of $\hat{\underline{U}}$ can be expected to be very flat around the correct solution $\underline{U}$ and, therefore, the process of recovering 3-D motion and structure will be very unstable and sensitive to noise.

To determine more precisely how these instability problems depend on factors related to the camera and the flow field, we will normalise the error field given in equation (3.15) to the noise level and thus, for each vector $\hat{\underline{U}}$, obtain a measure of a signal-to-noise ratio (SNR). Note that the error field is used as a 'signal' measure, since high values of this field reduce ambiguity. The probability that the vector $\hat{\underline{U}}$ will be selected as the correct solution is a decreasing function of the corresponding SNR values. Hence, if these values are small for a large set of translation axes, then instability in recovering 3-D information can be expected.

The noise in the flow field is assumed to be additive, its expectation is 0 and its standard deviation, in focal units, is

$$\sigma_f = \sigma_p \frac{2 \tan(fov/2)}{N}, \tag{3.18}$$

where the image contains $N \times N$ pixels and σ_p is the standard deviation in pixel units. To obtain an SNR measure we divide the error field by the square root of the sum of the second moments of the noise samples, which are assumed to be independent, in both axes. Thus, for each pixel (x, y) where a flow vector is defined,

$$snr(x,y) = \frac{\sqrt{(\Delta a_7 x^2 + \Delta a_8 xy)^2 + (\Delta a_7 xy + \Delta a_8 y^2)^2}}{2\sqrt{2}\sigma_p \tan(fov/2)/N}. \tag{3.19}$$

Employing the definition, in the beginning of this section, of the locality factor γ, x and y satisfy the inequalities:

$$x \le \gamma \tan(fov/2), \qquad y \le \gamma \tan(fov/2), \tag{3.20a,b}$$

and, therefore,

$$snr(x,y) \le \frac{N\gamma^2 \tan(fov/2)(|\Delta a_7| + |\Delta a_8|)}{2\sigma_p}. \tag{3.21}$$

Even when the SNR values are small, it may be possible to successfully recover the desired parameters, if there exist many flow vectors and the noise samples associated with them are independent. However, in many cases, especially if γ and N are small and the flow field is sparse, the number of flow vectors is small. In addition, if the flow field is dense, then usually the noise samples in neighboring pixels are highly correlated. This is the case, for example, if the noise is induced by a round-off error.

Before we can conclude this section we still have to deal with the depth constraint. This constraint is satisfied if, for any pixel (x, y) in the region $\mathcal{R}$, the estimated value of r/Z is positive, that is, the following inequality, derived from (2.6), holds:

$$\hat{l}_Z + \hat{l}_X x + \hat{l}_Y y > 0. \tag{3.22}$$

Substituting $\hat{l}_X$, $\hat{l}_Y$ and $\hat{l}_Z$ with the corresponding expressions in equations (3.14) and dividing by l_Z, we obtain the equivalent constraint:

$$\frac{U_Z}{\hat{U}_Z} + \frac{\epsilon_{XY}(\hat{U}_X s_Y - \hat{U}_Y s_X)}{\hat{U}_Z(\hat{U}_X^2 + \hat{U}_Y^2)} - \frac{(\hat{U}_X U_X + \hat{U}_Y U_Y)(s_X x + s_Y y) + \epsilon_{XY}(s_X y - s_Y x)}{\hat{U}_X^2 + \hat{U}_Y^2} > 0. \tag{3.23}$$

If the slopes s_X and s_Y are small and the region $\mathcal{R}$ is small, then, usually, the second and third terms in (3.23) are small and the inequality is satisfied by vectors $\hat{\underline{U}}$ in a large neighborhood of $\underline{U}$. Note that anyway these conditions are among those already specified as contributing to ambiguity.

To conclude, the following conditions contribute to ambiguity in recovering 3-D motion and structure parameters:

- The FOV is small.
- The locality factor γ is small.
- The planar surface is at most moderately slanted.
- The object is far away.
- The absolute value of the translation is small.
- The resolution of the image is coarse.
- The noise level (in pixels) is high.
- The flow field is sparse.
- The noise values in adjacent flow vectors are highly correlated.

3.2.2 An arbitrary surface

Referring to equation (2.5), the 'reciprocal depth' map, $1/Z$, can be generally represented by

$$\frac{1}{Z(x,y)} = k_Z + k_X x + k_Y y + \xi(x,y) \qquad (3.24)$$

where $\xi(x,y)$ is the difference between $1/Z$ and the approximating linear function $k_Z + k_X x + k_Y y$. Using this representation and the normalisation $\lambda = r\xi$, we can rewrite the flow field equations (2.2):

$$\alpha(x,y) = \alpha_\theta(x,y) + (U_X - xU_Z)\lambda(x,y) \qquad (3.25a)$$

and

$$\beta(x,y) = \beta_\theta(x,y) + (U_Y - yU_Z)\lambda(x,y), \qquad (3.25b)$$

where $(\alpha_\theta, \beta_\theta)$ is the Ψ transformation corresponding to the planar surface $k_X X + k_Y Y + k_Z Z = 1$.

Given $\underline{U}$, we can usually choose rotation parameter values $\hat{\Omega}$, and normalised plane parameter values $\hat{I}$, which maintain the correct seroth and first order components of the Ψ transformation. If, in addition, for each flow vector we choose the value of λ as the correct one, then the error field corresponding to these motion and structure parameters is

$$\begin{pmatrix} \Delta\alpha_\theta \\ \Delta\beta_\theta \end{pmatrix} + \begin{pmatrix} \Delta U_X - x\Delta U_Z \\ \Delta U_Y - y\Delta U_Z \end{pmatrix} \lambda(x,y), \qquad (3.26)$$

where $\Delta\alpha_\theta$ and $\Delta\beta_\theta$ are the errors associated with the planar surface (equation (3.15)) and $(\Delta U_X, \Delta U_Y, \Delta U_Z)$ is the error in the normalised translation vector. Therefore, we can expect instability in determining the 3-D motion and structure if, in addition to the conditions associated with planar surfaces, the function $1/Z$ can be reasonably approximated by a linear function, i.e., $\lambda(x,y)$ is small. Note that this condition means that the environmental surface can be relatively approximated by a planar surface, that is, the distance between the two surfaces is small relative to the distance from the sensor to the real surface. This approximation can usualy be improved as the FOV or the locality factor γ are reduced, unless there is a significant discontinuity in the depth map.

3.3 Examples

In this section we demonstrate the influence of three parameters on the degree of instability in recovering 3-D information from the flow field induced by a rigid motion of a planar surface. These parameters are the locality factor γ, the ratio r/d of the translation magnitude r to the distance d from the camera to the surface along the line of sight, and the slope s_X of the planar surface. The demonstration is based on several examples, where in each example a dense flow field is simulated. The technique presented in [ADI85a,b] for estimating 3-D motion and structure is then employed. In this technique, the search for 3-D motion parameters is based on a least-squares procedure which minimises the deviation between the given flow field and that predicted from the computed parameters. The minimisation algorithm employs an error measure corresponding to possible locations of the FOE in the image plane. For each hypothesised FOE, the optimal rotation parameters, the sign of the translation vector, and a related error value are computed. A minimal value of the resulting error function is determined, using a multi-resolution sampling technique.

The error function can be defined on the unit hemisphere $\mathcal{H} = \{\underline{U} : |\underline{U}| = 1,\ U_Z \geq 0\}$ which is isomorphic to the image plane. Employing a spherical coordinate system (r,ϕ,θ), where ϕ is the angle between the line of sight and the translation vector, and θ is the angle between the x-axis and the projection of the translation vector on the image plane, $\mathcal{H}$ can be represented by the set $\{(\phi,\theta) : 0 \leq \phi \leq 90°,\ 0° \leq \theta < 360°\}$. The angles (ϕ,θ) are used in Figures 3.2 to 3.8 as polar coordinates.

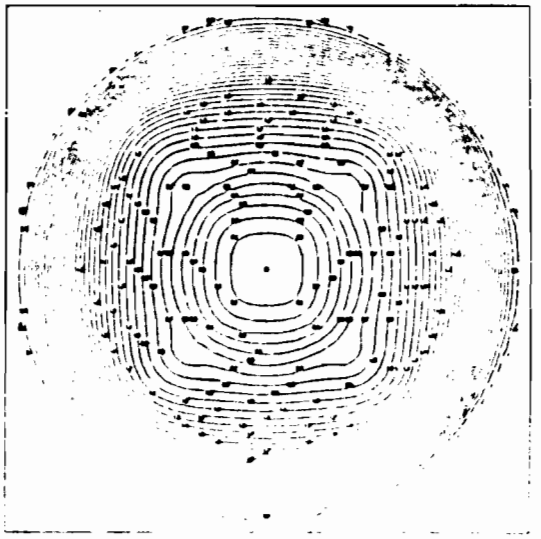


Figure 3.2: The error function in example 1. The surface is non-slanted, $r/d = 0.1$, and $\gamma = 1$.

Note that ϕ ranges from 0° at the center up to 90° at the boundary.

The sharpness of the error function around the correct value determines the sensitivity to noise in estimating the translation axis and, therefore, also in estimating the rotation parameters and the environmental structure. In all the examples the FOV is 30°, the number of pixels is 128×128, the camera translation is $\mathbf{T} = (0,0,10)$ and the rotation is $(0,0,0)$. In the first three cases the surface, defined by the equation $Z = 100$, is parallel to the image plane. The locality factor γ, on the other hand, is different in each of these experiments: 1 in the first, 1/2 in the second and 1/4 in the third. The contour maps in Figures 3.2, 3.3 and 3.4 show the drastic change in the sharpness of the corresponding error functions. The contours are labeled by the corresponding error values, given in pixels, and the correct solution is marked by a black dot.

In examples 4 to 7 we choose again $\gamma = 1$, but the planar surface is varied. In examples 4 and 5 the surface is still parallel to the image plane, but its distance from the camera is 200 in example 4, and 400 in example 5. Thus, the influence of the relative translation magnitude r/d is demonstrated by examples 3, 4 and 5, in which r/d is 0.1, 0.05 and 0.025, respectively. The results in Figures 3.2, 3.5 and 3.6 clearly show that smaller values of r/d are associated with higher levels of ambiguity.

In examples 6 and 7 the distance d is kept at 100, but the surfaces, defined respectively by $Z = 100 + 0.414X$ and $Z = 100 + X$, are slanted: 22.5° in example 6 and 45° in example 7. Figures 3.2, 3.7 and 3.8 show that the error function becomes sharper as the surface becomes more slanted. The basic reason for this relation is the depth variation associated with slanted surfaces. This variation helps in resolving the ambiguity in distinguishing between the translational and rotational components of the flow field since the first component is affected by variations in depth, while the second component is independent of the depth values.

Note the second accurate solution in Figures 3.7 and 3.8, which corresponds, according to equations developed in [WAX83], to a situation where the surface is non-slanted but the translatory motion is not along the line of sight; instead, the motion deviates by 22.5° and 45°, respectively, from this line. The relative translation along the line of sight, that is, T_Z/d, is still 0.1 for these alternative solutions. Since in these cases the translational motion along the X-axis is non-zero, the ratio r/d is larger than 0.1, specifically 0.1082 in experiment 6, and 0.1414 in experiment 7.

The solution in experiment 1 and the alternative solutions in experiments 6 and 7 correspond to situations where the surface is parallel to the image plane. Yet, there is a large difference in the sharpness of the error functions around these solutions. This difference may partly be due to the change in r/d. The second factor which apparently influences the degree of ambiguity in these cases is the deviation between the line of sight and the translation axis. At

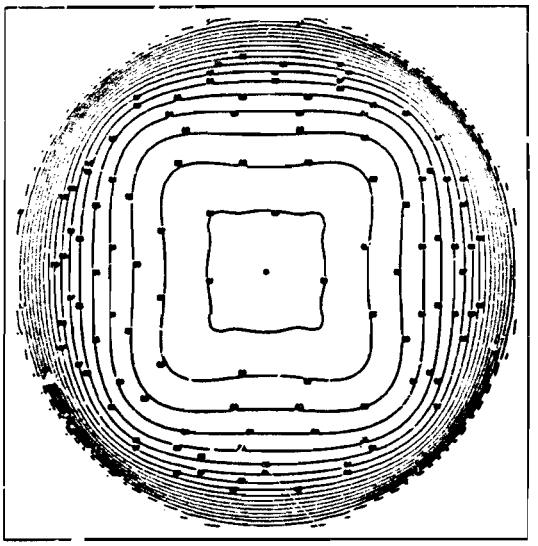


Figure 3.3: The error function in example 2. The surface is non-slanted, $r/d = 0.1$, and $\gamma = 0.5$.

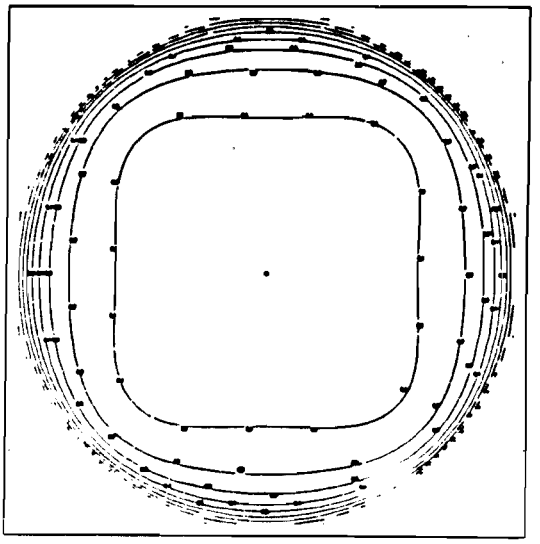


Figure 3.4: The error function in example 3. The surface is non-slanted, $r/d = 0.1$, and $\gamma = 0.25$.

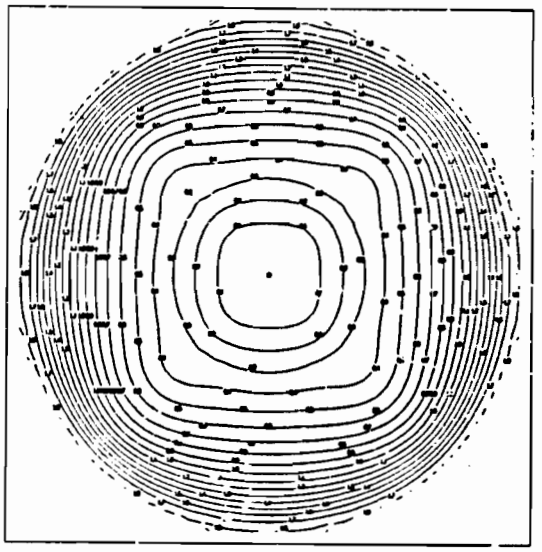


Figure 3.5: The error function in example 4. The surface is non-slanted, $r/d = 0.05$, and $\gamma = 1$.

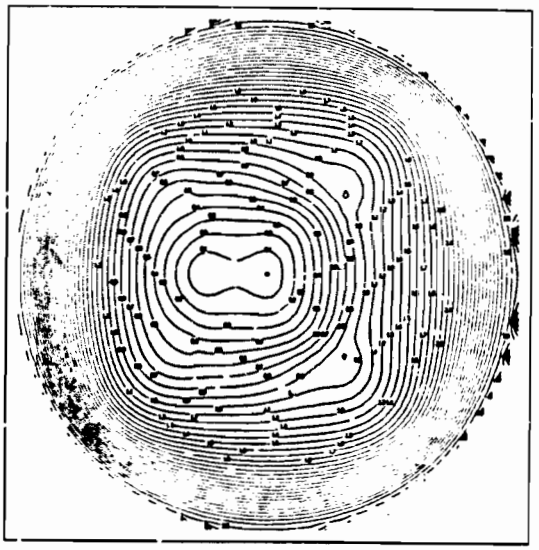


Figure 3.7: The error function in example 6. The surface is slanted (22.5°), $r/d = 0.1$, and $\gamma = 1$. Note the second solution which corresponds to a non-slanted surface and a translation not along the line of sight (the angle between the translation vector and the line of sight is 22.5°).

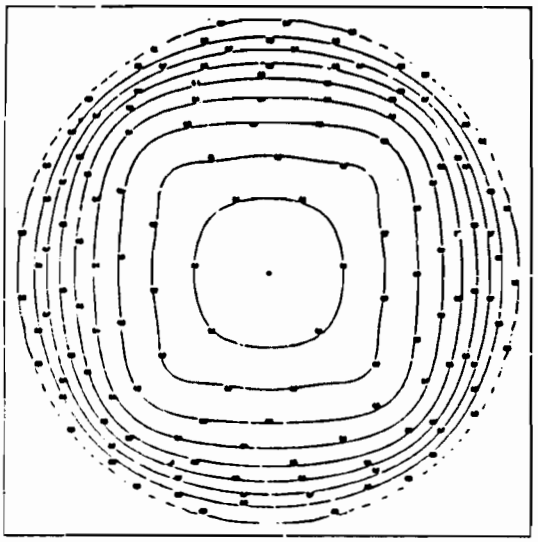


Figure 3.6: The error function in example 5. The surface is non-slanted, $r/d = 0.025$, and $\gamma = 1$.

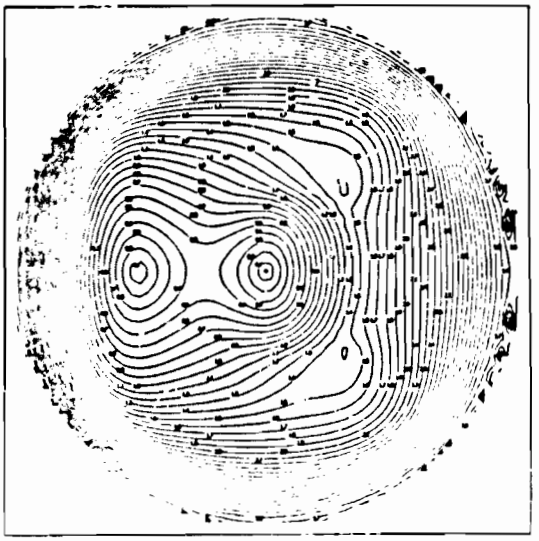


Figure 3.8: The error function in example 7. The surface is slanted (45°), $r/d = 0.1$, and $\gamma = 1$. The second solution corresponds again to a non-slanted surface and a translation not along the line of sight (this time, the angle between the translation vector and the line of sight is 45°).

least in the case of a non-slanted planar surface, it seems that the instability is reduced as the translation vector increasingly deviates from the line of sight.

3.4 Constraints and Parameters which Can be Determined

In ambiguous situations, when the surface can be relatively approximated by a plane, we can still recover useful information in terms of partial constraints on the motion and structure parameters. Usually, the coefficients of the 0th and 1st order components of the flow field, that is, the coefficients $a_1, \ldots, a_6$ of the Ψ transformation (see equations (2.8a-f)), can be reliably estimated. Integration of these constraints over a time sequence of flow fields may, eventually, resolve the ambiguity and result in a unique interpretation.

If a planar patch is independently moving and the camera is stationary, then the ambiguity is, at least partially, the result of using a *camera coordinate system*. In this coordinate system a_1 and a_4 are sums of the X and Y translations (normalised by the distance d from the camera to the object along the line of sight) and rotations. It may be very difficult, however, to determine the correct decomposition to the rotational and translational components. On the other hand, it is possible to define an *'object coordinate system'* which is parallel to the camera coordinate system, but its center is shifted to the surface along the line of sight. In this coordinate system a_1 and a_4 are, respectively, the X and Y translations normalised by d. Hence, *these* normalised translations can be reliably recovered.

Let us now examine the situation where at least one of the two following conditions is satisfied: (a) the translation is along the line of sight, that is, $U_X = U_Y = 0$; (b) the surface can be relatively approximated by a planar surface parallel to the image plane, that is, $l_X = l_Y = 0$. Note that this situation is very common in real scenes. Employing equations (2.8c,e), $a_3 = -a_5$ in this case, and Ω_Z can be estimated by $(a_5 - a_3)/2$. In addition, $a_2 = a_6$ and T_Z, normalized by the distance to the object along the line of sight, can be estimated by $(a_2 + a_6)/2$. In this situation, this is the inverse of the *time-to-collision* and, therefore, we can usually obtain a reasonably accurate estimate of this important parameter, even when ambiguity in recovering 3-D information does exist.

In order to show how the situation discussed above can be detected, we will prove that when $a_3 = -a_5$ and $a_2 = a_6$, then $U_X = U_Y = 0$ and/or $l_X = l_Y = 0$. That is, the first equalities are not only necessary but also sufficient conditions for the latter ones. To prove this, notice that the equalities $a_3 = -a_5$ and $a_2 = a_6$, combined with equations (2.8b,c,e,f), lead to the equalities:

$$l_Y U_X = -l_X U_Y, \qquad (3.27a)$$

$$l_X U_X = l_Y U_Y. \qquad (3.27b)$$

Assuming that U_X, U_Y, l_X and l_Y are all non-zero, we can divide each side of the first equation with the corresponding side of the second equation, and thus obtain $l_Y/l_X = -l_X/l_Y$ which leads to a contradiction: $(l_Y/l_X)^2 = -1$. Therefore, at least one of the quantities U_X, U_Y, l_X, l_Y must be 0. Suppose now that $U_X = 0$; examining equations (3.27), it follows that $l_X U_Y = 0$ and $l_Y U_Y = 0$ and, therefore, $U_Y = 0$ and/or $l_X = l_Y = 0$. Similarly, each of the conditions $U_Y = 0$, $l_X = 0$, $l_Y = 0$ leads to the desired result.

Another approach which may be taken in order to deal with instability in recovering motion parameters is based on representing possible values of these parameters by a probabilistic distribution function. Such a function can be defined, for example, on the unit hemisphere $\mathcal{H}$, using the computed values of the corresponding error field in [ADI85a,b].

4. Ambiguity in Grouping Flow Vectors into Sets Corresponding to Rigidly Moving Objects

4.1 Introductory Discussion

We demonstrated in Section 3 that in some situations there exists a large set of motion parameters which, assuming the presence of noise, are consistent with the flow field generated by a rigidly moving object. Suppose now that two independently moving objects are given. If the two corresponding solution sets of motion parameters are large, then the possibility that these sets intersect each other is not negligible. Such an intersection corresponds to 3-D motion parameters which are consistent with both objects. In this case the optical flows can be interpreted as resulting from one rigidly moving object. Note that, to the best of our knowledge, this ambiguity has not been addressed yet in the literature.

To demonstrate the possibility of ambiguity in decomposing the flow field into sets corresponding to rigid objects, we wish to show that there exist non-trivial situations in which we can find motion parameters compatible with the flows generated by two independently moving objects. If the surfaces of the objects can be relatively approximated by planes, then, following Section 3.2, we can examine this possibility of ambiguity by trying to compute motion and structure parameters which are consistent with the coefficients $a_1, \ldots, a_6$ of the associated Ψ transformations. The resulting errors in the second-order components of the optical flows will be small if, for example, the field of view is small enough. Since the six motion parameters $\underline{U}$ and $\underline{\Omega}$ should be the same for both objects but the three structure parameters can be different, we obtain, including the constraint $U_X^2 + U_Y^2 + U_Z^2 = 1$, 13 equations with 12 unknowns. It is reasonable to expect that in many situations these equations do have a solution.

4.2 Analysis of a Specific Case

Continuing the introductory discussion, let us now examine, as we did in Section 3.4, the common situation where, for each of the two objects, at least one of the following conditions (not necessarily the same) must be satisfied: the translation is along the line of sight, or the surface can be relatively approximated by a planar surface parallel to the image plane. In such a situation $U_X = U_Y = 0$ or $l_X = l_Y = 0$, and $U'_X = U'_Y = 0$ or $l'_X = l'_Y = 0$, where the parameters associated with the second object are marked by the symbol "'". In addition, let us assume that $\Omega_Z = \Omega'_Z$ and that the signs of U_Z and U'_Z are the same, where $U_Z = 0$ if and only if $U'_Z = 0$. Finally, to guarantee a solution, it is assumed that if $U_Z \neq 0$ and $l_Z U_Z = l'_Z U'_Z$, then $\Omega_Y + l_Z U_X = \Omega'_Y + l'_Z U'_X$ and $-\Omega_X + l_Z U_Y = -\Omega'_X + l'_Z U'_Y$.

Employing equations (2.8a) to (2.8f), we can obtain the following equations, related to the first object, where the unknowns are denoted by the symbol "^":

$$\hat{\Omega}_Y + \hat{l}_Z \hat{U}_X = a_1 \ (= \Omega_Y + l_Z U_X), \tag{4.1a}$$

$$\hat{l}_X \hat{U}_X - \hat{l}_Z \hat{U}_Z = a_2 \ (= -l_Z U_Z), \tag{4.1b}$$

$$-\hat{\Omega}_Z + \hat{l}_Y \hat{U}_X = a_3 \ (= -\Omega_Z), \tag{4.1c}$$

$$-\hat{\Omega}_X + \hat{l}_Z \hat{U}_Y = a_4 \ (= -\Omega_X + l_Z U_Y), \tag{4.1d}$$

$$\hat{\Omega}_Z + \hat{l}_X \hat{U}_Y = a_5 \ (= \Omega_Z) \tag{4.1e}$$

and

$$\hat{l}_Y \hat{U}_Y - \hat{l}_Z \hat{U}_Z = a_6 \ (= -l_Z U_Z). \tag{4.1f}$$

A similar set of equations can be obtained for the second object.

Let us start the solution process by choosing $\hat{\Omega}_Z = \Omega_Z$, $\hat{l}_X = \hat{l}_Y = 0$ and $\hat{l}'_X = \hat{l}'_Y = 0$, thus satisfying equations (4.1c) and (4.1e), as well as the corresponding equations associated with the second object. We proceed by examining the case $U_Z \neq 0$, in which we constrain $\hat{U}_Z$ to be non-zero and to have the same sign as the sign of U_Z. Thus, from (4.1b) and (4.1f) we can obtain

$$\hat{l}_Z = l_Z \frac{U_Z}{\hat{U}_Z}. \tag{4.2}$$

Substituting this expression in (4.1a) and (4.1d) yields

$$\hat{\Omega}_Y + l_Z U_Z \hat{m}_x = a_1 \tag{4.3a}$$

and

$$-\hat{\Omega}_X + l_Z U_Z \hat{m}_y = a_4, \tag{4.3b}$$

where $(\hat{m}_x, \hat{m}_y) = (\hat{U}_X/\hat{U}_Z, \hat{U}_Y/\hat{U}_Z)$ is the corresponding FOE. Similarly, we can obtain the following equations, corresponding to the second object:

$$\hat{\Omega}_Y + l'_Z U'_Z \hat{m}_x = a'_1 \tag{4.4a}$$

and

$$-\hat{\Omega}_X + l'_Z U'_Z \hat{m}_y = a'_4. \tag{4.4b}$$

Combining (4.3) with (4.4) yields

$$a_1 - l_Z U_Z \hat{m}_x = a'_1 - l'_Z U'_Z \hat{m}_x \tag{4.5a}$$

and

$$a_4 - l_Z U_Z \hat{m}_y = a'_4 - l'_Z U'_Z \hat{m}_y. \tag{4.5b}$$

If $l_Z U_Z = l'_Z U'_Z$, then, according to our assumptions, $a_1 = a'_1$ and $a_4 = a'_4$, and, therefore, we can choose arbitrary values of $\hat{m}_x$ and $\hat{m}_y$. Otherwise,

$$\hat{m}_x = \frac{a'_1 - a_1}{l'_Z U'_Z - l_Z U_Z} \tag{4.6a}$$

and

$$\hat{m}_y = \frac{a'_4 - a_4}{l'_Z U'_Z - l_Z U_Z}. \tag{4.6b}$$

The values of $\hat{\Omega}_X$ and $\hat{\Omega}_Y$ can now be computed from equations (4.3) or (4.4).

Let us now examine the complementary case where $U_Z = 0$. In this case, to satisfy equations (4.1b) and (4.1f), we choose $\hat{U}_Z = 0$. Combining equations (4.1a) and (4.1d) with the corresponding equations associated with the second object yields

$$a_1 - \hat{l}_Z \hat{U}_X = a'_1 - \hat{l}'_Z \hat{U}_X \tag{4.7a}$$

and

$$a_4 - \hat{l}_Z \hat{U}_Y = a'_4 - \hat{l}'_Z \hat{U}_Y. \tag{4.7b}$$

Therefore,

$$(\hat{l}'_Z - \hat{l}_Z)\hat{U}_X = a'_1 - a_1 \tag{4.8a}$$

and

$$(\hat{l}'_Z - \hat{l}_Z)\hat{U}_Y = a'_4 - a_4. \tag{4.8b}$$

Thus, since $\hat{U}_X^2 + \hat{U}_Y^2 = 1$,

$$\hat{l}'_Z - \hat{l}_Z = \pm\sqrt{(a'_1 - a_1)^2 + (a'_4 - a_4)^2}. \tag{4.9}$$

If $a_1 = a'_1$ and $a_4 = a'_4$, then $\hat{l}_Z = \hat{l}'_Z$, and any $\hat{U}_X$ and $\hat{U}_Y$ which satisfy $\hat{U}_X^2 + \hat{U}_Y^2 = 1$ are legitimate solutions. Otherwise, that is, if $a_1 \neq a'_1$ or $a_4 \neq a'_4$, then

$$\hat{U}_X = \frac{a'_1 - a_1}{\pm\sqrt{(a'_1 - a_1)^2 + (a'_4 - a_4)^2}} \tag{4.10a}$$

and

$$\hat{U}_Y = \frac{a'_4 - a_4}{\pm\sqrt{(a'_1 - a_1)^2 + (a'_4 - a_4)^2}}. \tag{4.10b}$$

To finish the solution process, we should choose $\hat{l}_Z$ and $\hat{l}'_Z$ which satisfy the constraint (4.9) and, then, using equations

(4.1a) and (4.1d), determine the values of $\hat{\Omega}_X$ and $\hat{\Omega}_Y$. It is optimal to select the values of $\hat{l}_Z$ and $\hat{l}'_Z$ such that the resulting errors in the coefficients a_7 and a_8 of the Ψ transformations will be minimal.

4.3 An Example

In order to demonstrate how different motions can be interpreted as one rigid motion, let us examine the case where two planar patches, parallel to the image plane, are independently translating. Both translations are assumed to be parallel to the image plane, but one object is translating in parallel to the X-axis generating flow values of $(-0.04, 0)$ (in focal units), and the second object is translating in parallel to the Y-axis generating flow values of $(0, 0.03)$ (see Figure 4.1). Note that $a_1 = -0.04$, $a'_4 = 0.03$ and the other coefficients of the Ψ transformations associated with the objects are 0.

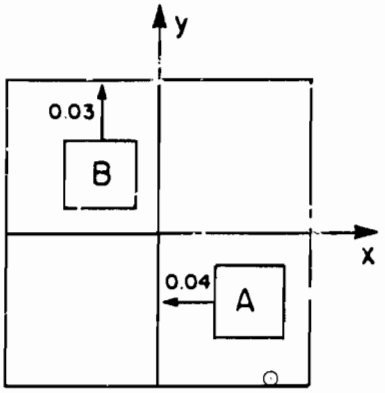


Figure 4.1: The optical flows induced by the translation of two objects.

We wish to recover motion parameters $\hat{\Omega}$ and $\hat{\Omega}'$ which are compatible with both sets of flow vectors and with structure parameters $\hat{l}$ and $\hat{l}'$ corresponding, respectively, to the first and second object. Employing the results in Section 4.2 for the case $U_Z = 0$ (equations (4.10)), we have $\hat{\Omega} = (\pm 0.8, \pm 0.6, 0)$, $\hat{\Omega}_Z = 0$, $\hat{l}_X = \hat{l}_Y = 0$, $\hat{l}'_X = \hat{l}'_Y = 0$ and $\hat{l}'_Z - \hat{l}_Z = \pm 0.05$. In addition, using equations (4.1c) and (4.1d), $\hat{\Omega}_X = \pm 0.6\hat{l}_Z$ and $\hat{\Omega}_Y = -0.04 \mp 0.8\hat{l}_Z$.

Since $\hat{\Omega}_X$ and $\hat{\Omega}_Y$ are exactly the errors in the coefficients a_7 and a_8 of the Ψ transformations, we wish to minimise

$$\chi(\hat{l}_Z) \stackrel{\text{def}}{=} \hat{\Omega}_X^2 + \hat{\Omega}_Y^2 = 0.36\hat{l}_Z^2 + (0.04 \pm 0.8\hat{l}_Z)^2. \quad (4.11)$$

Let us now distinguish between the cases $\hat{l}'_Z > \hat{l}_Z$ and $\hat{l}'_Z < \hat{l}_Z$. In the first case $\hat{\Omega} = (0.8, 0.6, 0)$ and the first derivative of $\chi(\hat{l}_Z)$ is $2\hat{l}_Z + 0.064$. Since $\hat{l}_Z$ is constrained to be positive, the minimum of $\chi(\hat{l}_Z)$, obtained for $\hat{l}_Z = 0$, is, in this case, 0.0016. Note that $\hat{l}_Z = 0$ means that the object is at infinity, which is, of course, unrealistic; however, taking a sufficiently large distance of the object from the camera, $\hat{l}_Z$ can be arbitrarily close to 0.

In the second case $\hat{\Omega} = (-0.8, -0.6, 0)$, the derivative of $\chi(\hat{l}_Z)$ is $2\hat{l}_Z - 0.064$ and $\hat{l}_Z$ should be at least 0.05. Hence, the minimum of $\chi(\hat{l}_Z)$, achieved for $\hat{l}_Z = 0.05$, is, in this case, 0.0009. The optimal solution is, therefore, $\hat{\Omega} = (-0.8, -0.6, 0)$, $\hat{l} = (0, 0, 0.05)$, $\hat{\Omega}' = (0, 0, 0)$ and $\hat{l}' = (-0.03, 0, 0)$. Assuming small second order terms of the rotational component, this solution can be graphically represented by Figure 4.2.

Since $\hat{\Omega}_Y = 0$, there is no error in a_7; on the other hand, there is an error in a_8 which is $-\hat{\Omega}_X = 0.03$. The corresponding discrepancy between the correct flow field and that predicted from the above parameters is small if the FOV is small, or both the size of the objects and their distance from the line of sight are small relative to their distance from the camera.

4.4 Concluding Remarks on the Rigidity Assumption

We have just shown that a rigidity assumption, similar to the one proposed in [ULL79], is not appropriate when the flow field is noisy, that is, the consistency of a set of flow vectors with the same 3-D motion parameters does not reasonably guarantee that they are really induced by one rigidly moving object. Observing, in addition, that almost any set which contains less than 5 flow vectors is consistent with some Ψ transformation, we propose a modified assumption: *a set of at least 5 adjacent flow vectors, which are compatible, up to the estimated noise level, with a rigid motion of a planar patch, will be assumed to be induced by one rigidly moving object.* This assumption has been successfully applied [ADI85a,b] to segment noisy flow fields.

In some situations, however, the consistency of two sets of flow vectors with the same motion parameters is still a strong evidence for the hypothesis that these sets are generated by one rigidly moving object. This is the case, for example, when accurate motion parameters can be separately recovered for each set. In such a situation, similarity of the results is not likely to be accidental.

Nevertheless, in general we still must accept the possibility of ambiguity in grouping flow vectors into sets corresponding to rigidly moving objects. Hence, the interpretation of the flow field should result in a *set* of possible decompositions, rather than only one decomposition. Each hypothesised object can be assigned a probability value, based on the number of segments composing the object's flow and on the degree of ambiguity in separately recovering the motion parameters associated with each of them.

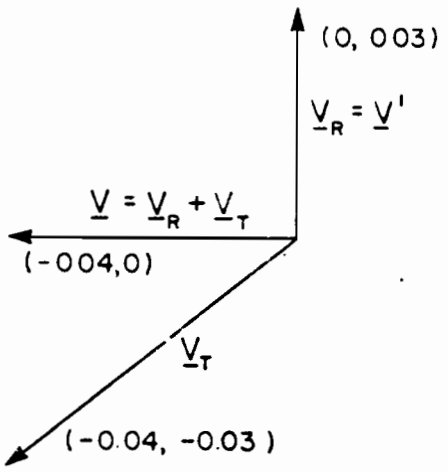


Figure 4.2: Graphical representation of the optimal solution. The flow vectors corresponding to the first and second object are denoted by $\underline{v}$ and $\underline{v}'$, respectively. The rotational component of the optimal solution is $\underline{v}_R$, while the translational component corresponding to the first object is $\underline{v}_T$ and the translational component corresponding to the second object is (0,0).

5. Conclusions

We have characterised and demonstrated situations in which there exists an inherent ambiguity in the interpretation of noisy flow fields. The first ambiguity is in recovering the motion parameters from a noisy flow field generated by a rigid motion. We found that if the field of view corresponding to the region containing the interpreted flow field is small, and the depth variation and translation magnitude are small relative to the distance of the object from the camera, then the determination of the 3-D motion and structure can be expected to be very sensitive to noise and, in the presence of a realistic level of noise, practically impossible. We experimentally found that there is also a relation between the location of the FOE and the degree of ambiguity. This relation should be mathematically investigated in future research.

The second ambiguity is in the decomposition of the flow field into sets corresponding to independently moving objects. We found that the rigidity assumption is not appropriate for noisy flow fields, that is, the consistency of a set of flow vectors with the same motion parameters, up to the estimated noise level, does not reasonably guarantee that they are really induced by one rigid motion. As an alternative to this assumption, it is assumed in [ADI85a,b] that a connected set of flow vectors, which is consistent with a rigid motion of a *planar* surface, is induced by a single rigid motion. This assumption is weaker than the first version of the rigidity assumption in the sense that it can only be applied in more restricted situations and, therefore, it is more likely to be correct.

The results of the ambiguity analysis can be used when the effectiveness of motion algorithms is evaluated for real-world tasks. They can help to decide which algorithm to choose, and in what situations this algorithm can be expected to be effective.

Constraints and parameters which can be extracted, even in ambiguous situations, were also introduced. Integration of such partial information over a time sequence of flow fields may, eventually, resolve the ambiguity and result in a unique interpretation. In addition, combining this information with other knowledge sources (e.g., a fiber optic rotation sensor [LAW84]) can be considered.

Recovering motion and structure of independently moving objects may be particularly difficult, as was demonstrated by the flat error surfaces obtained for such objects in the second and fifth experiments in [ADI85b]. In general, ambiguity in recovering 3-D motion and structure of independently moving objects can be expected, since the effective field of view and the ratio of the depth variation to the distance between the object and the camera are usually small. Furthermore, additional information from other knowledge sources may be hard to acquire. Therefore, the possibility of partially resolving the ambiguity in such a case, by using an *object* coordinate system, is especially interesting and should be investigated in future research.

Acknowledgements

I would like to thank Ed Riseman and Al Hanson for useful comments for improving this paper. I am also indebted to Brian Burns for his help in preparing the contour maps in Figures 3.2 to 3.8.

References

[ADI85a] G. Adiv, *Determining 3-D Motion and Structure from Optical Flow Generated by Several Moving Objects*, IEEE Trans. Pattern Anal. Machine Intell., vol PAMI-7, pp. 384–401, July 1985.

[ADI85b] G. Adiv, *Interpreting Optical Flow*, Ph.D. Dissertation, Computer and Information Science Dept., Univ. of Mass., 1985.

[FAN83a] J.-Q. Fang and T.S. Huang, *Solving Three Dimensional Small-Rotation Motion Equations*, in Proc. IEEE Conf. Computer Vision and Pattern Recognition, Washington, D.C., 1983, pp. 253-258.

[FAN83b] J.-Q. Fang and T.S. Huang, *Estimating 3-D Movement of a Rigid Object: Experimental Results*, in Proc. Int. Joint Conf. Artificial Intell., Karlsruhe, Germany, 1983, pp. 1035-1037.

[HAN78] A. Hanson and E. Riseman (Eds.), "Computer Vision Systems", Academic Press Inc., New York, NY, 1978, pp. 303-334.

[LAW84] D.T. Lawton, *Processing Dynamic Image Sequences from a Moving Sensor*, Ph.D. Dissertation (TR 84-05), Computer and Information Science Dept., Univ. of Mass., 1984.

[LON80] H.C. Longuet-Higgins and K. Prazdny, *The interpretation of a Moving Retinal Image*, Proc. Roy. Soc. Lond., B, vol. 208, pp. 385-397, July 1980.

[LON81] H.C. Longuet-Higgins, *A Computer Algorithm for Reconstructing a Scene from Two Projections*, Nature, vol. 293, pp. 133-135, Sep. 1981.

[PRA80] K. Prazdny, *Egomotion and Relative Depth Map from Optical Flow*, Biol. Cybernetics, vol. 36, pp. 87-102, 1980.

[RAL65] A. Ralston, "A First Course in Numerical Analysis", McGraw-Hill, New York, NY, 1965.

[RIE83] J.H. Rieger and D.T. Lawton, *Determining the Instantaneous Axis of Translation from Optic Flow Generated by Arbitrary Sensor Motion*, in Proc. Workshop Motion: Representation and Perception, Toronto, Canada, 1983, pp. 33-41.

[ROA80] J.W. Roach and J.K. Aggarwal, *Determining the Movement of Objects from a Sequence of Images*, IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-2, pp. 554-562, Nov. 1980.

[TSA84] R.Y. Tsai and T.S. Huang, *Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces*, IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-6, pp. 13-27, Jan. 1984.

[ULL79] S. Ullman, "The Interpretation of Visual Motion", MIT Press, Cambridge, Mass., 1979.

[ULL81] S. Ullman, *Analysis of Visual Motion by Biological and Computer Systems*, Computer, vol. 14, pp. 57-69, Aug. 1981.

[WAX83] A.M. Waxman and S. Ullman, *Surface Structure and 3-D Motion from Image Flow: A Kinematic Approach*, CAR-TR-24, Center for Automation Research, Univ. of Maryland, 1983.

GC970145-44

Refinement Of Environmental Depth Maps Over Multiple Frames

Seraj Bharwani, Edward Riseman, Allen Hanson
Computer and Information Sciences Departement
University of Massachusetts
Amherst, Ma 01003

Abstract

In this paper we examine the task of constructing a reliable depth map of the environment from a sequence of images obtained from a camera undergoing translational motion. Even when the motion of the camera is known, local ambiguities occur in the matching of features from one frame to the next leading to ambiguity in the recovery of environmental depth. This paper first examines the sources of error in computing depth and then proposes mechanisms to obtain coarse estimates of depth, predict displacements and refine the depth map for selected feature points. The technique iteratively improves the accuracy of the depth estimates over a sequence of frames, while maintaining constant computational limits on processing between frames. Both start-up and updating strategies follow as part of a hierarchical spatial and temporal processing paradigm. The results of a preliminary implementation are presented and discussed.

1 Introduction

The process of recovering structure from general motion has been studied by several researchers. Some have used point correspondence and the rigidity assumption to recover the 3-D structure [Ullm83,Tsai84], some have combined the problem of measuring motion and of recovering structure using global/local search techniques[Lawt84], while others rely on the properties of optical flow to recover surface structure [Waxm83,Adiv85]. A problem shared by each of these is that the process requires more computation than that which is available at frame rate for real time implementation. In addition, almost all of the work has been restricted to processing only two image frames.

In this paper we are concerned with the problem of recovering depth maps over multiple frames for a known translational motion of a camera. The focus of expansion (FOE) is the point where the axis of camera translation intersects the image plane. The displacement path is defined as the path along the line in the image connecting the point and the FOE. For a camera in translation, *points in the image are displaced along the displacement path.* Given the axis of translation of the camera, the depth of a point can be determined from the position of the point and the extent of its displacement relative to the focus of expansion (FOE) or the focus of contraction (FOC). This relation is given by

$$\frac{D}{d} = \frac{Z}{T_Z} \tag{1}$$

where T_Z is the displacement of the camera along the Z axis from time t to time $t+1$, Z is the Z-coordinate (called the *depth* of the point relative to the camera) of an environmental point at time $t+1$, D is the distance of the corresponding image point from the FOE or FOC at time t, and d is the magnitude of the displacement of the image point from time t to $t+1$[Ball82]. Refer to Figure 1 for details on the geometry of the camera and the relation of the camera coordinate system (Y, Z) to the world coordinate or the global coordinate system $(\acute{Y}, \acute{Z})$. The depth of a point can thus be computed from equation (1) if the magnitude of the camera translation is known for the time interval between any two frames; otherwise, only relative depth can be computed as a function of camera translation.

For schemes that use correlation for finding correspondences between features, a common cause of inconsistencies in the recovery of environmental depth of the surface patch associated with the feature is the occurrence of no matches for features perhaps due to occlusion or ambiguous matches resulting from an inability to sufficiently localize the search interval for finding correspondence in subsequent frames. Additionally, there is a limit on the accuracy with which displacements can be measured given the limit on the amount of computation available to carry out interpolation for obtaining subpixel accuracy. It is our goal to first identify the source of such difficulties in obtaining accurate depth maps. Later we discuss our approach to the computation of depth maps which is based on the assumption that accurate construction of environmental depth is a gradual process requiring only coarse estimates of depth in the early stages of processing (i.e. the "start-up" phase

[1]This research was supported by the US Army ETL under contract number DACA76-85-C-0008 and by Defense Advanced Research Projects Agency under contract number N00014-82-K-0464

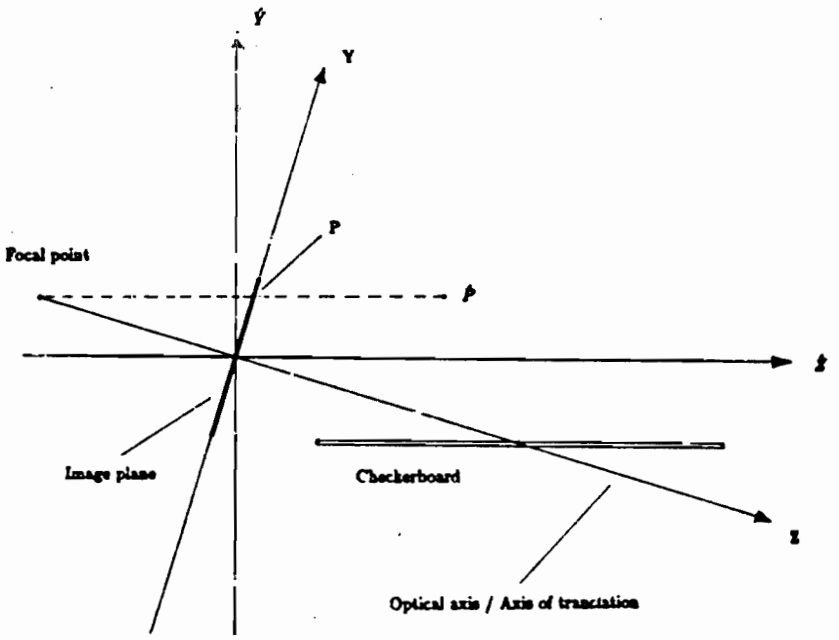


Figure 1: The camera geometry and the coordinate system. Y,Z is the camera-centered coordinate system and $\acute{Y}$, $\acute{Z}$ is the world coordinate system.

where no depth maps initially exist). These estimates can then be iteratively updated over several frames using finer resolution matching in a prediction and refinement scheme.

2 Representation

Currently our representation of the world map involves the representation of depth of a set of feature points in the image. Good candidates for feature points are those that are likely to have a unique match such as points of high curvature along image contours, etc. Points with such unique local characteristics could be selected by applying an interest operator[Kitc80,Mora77]. Once the features have been selected, they are tagged with a symbolic identifier. The key information, however, is that each feature point carries with it a list that contains the following information:

- the x and y coordinate of the point in the image;
- the depth estimate of the point from the previous time step
- the expected accuracy of the depth estimate from the previous time step

In some cases it may be desirable to maintain a longer processing history. In this case, the information attached to each feature point becomes a list of lists.

The depth computation in our framework is always performed with respect to the camera-centered coordinate system. Since the camera translates T_Z through three-dimensional space, all predictions from the previous time step $t-1$ have to be transformed to be consistent with the origin of the current coordinate system at time t. Similarly, any comparisons in depth for verification would also require a transformation. If the camera continues to move at constant speed then the measurements from N frames previously will need to have their z coordinate modified by $N \cdot T_Z$ in order to compare them.

3 Errors In Depth Measurement

Errors in depth measurement from translational motion arise from several sources and the effect of each must be understood. A brief list of such sources is given below:

- Errors due to incorrect match along a displacement path can produce arbitrary image displacement error and therefore arbitrarily large depth error;
- No match (i.e. lack of an acceptable match) might occur due to
  1. an insufficiently large search area along the displacement path for the match;
  2. occlusion of the feature in the next frame;
  3. noise, highlights, shadows and surface distortion.
- An error in the FOE which causes an incorrect dis-

placement path for most points in the image (this source of error is assumed to be minimal in this paper).

- The accuracy in measured displacements is limited by the discrete nature of the matching/correlation (search) process; i.e., only a discrete set of window positions are checked for possible matches. The search interval is the distance between these points and the displacement accuracy corresponds to this interval as shown in Figure 2.

The prediction-guided depth refinement process that we discuss later in this paper focuses upon some of these types of error. In particular, the accuracy in the computation of displacement between one pair of frames can be used to limit the interval within which to search for the displaced point in subsequent frames.

The relationship between the displacement accuracy and the depth accuracy can be obtained as follows. Let δd be the magnitude of the accuracy in measuring the displacement of the point in the image and let δZ be the corresponding accuracy in depth of the point. Since

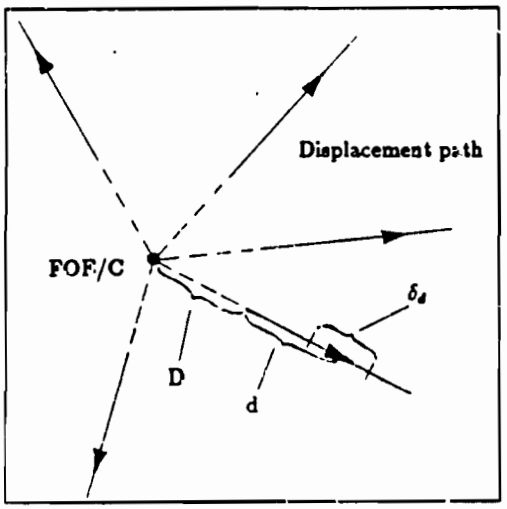


Figure 2: Displacement accuracy δ_d along the displacement path.

$$Z_t = T_Z \cdot D_t / d_t \tag{2}$$

where Z_t and D_t are the depth and the distance of the point from the FOE respectively and d_t is the displacement of the point from time $t-1$ to time t. By Differentiating (2) with respect to d_t results in:

$$|\delta Z_t| = |\frac{D_t \cdot T_Z}{d_t^2}| \cdot |\delta d_t| \tag{3}$$

For a given d_t, we can substitute for it in (3) to obtain

$$|\delta Z_t| = |\frac{Z_t^2}{D_t \cdot T_Z}| \cdot |\delta d_t| \tag{4}$$

Hence, for obtaining accurate depths, measuring displacements accurately becomes more critical the farther the point is from the camera. In actual situations, one could expect the camera to be in constant motion, with image frames acquired at some fixed temporal interval. In these situations, the process of computing environmental depths can be viewed as an iterative refinement process, characterised by measurement of depth followed by prediction of image displacement and refinement of displacement and depth estimates. Since we have an estimate of the accuracy in computed depth at time t, we can compute the expected accuracy in displacement at $t+1$ from

$$|\delta d_{t+1}| = |\frac{D_{t+1} \cdot T_Z}{\hat{Z}_{t+1}^2}| \cdot |\delta \hat{Z}_{t+1}| \tag{5}$$

where $\hat{Z}_{t+1}$ and $\delta \hat{Z}_{t+1}$ are the expected depth and the expected accuracy in depth respectively at time $t+1$. The expected depth at time $t+1$ is given by

$$\hat{Z}_{t+1} = Z_t - T_Z \tag{6}$$

so that

$$|\delta \hat{Z}_{t+1}| = |\delta Z_t|$$

if we assume that the translation of the camera T_Z is known accurately.

4 Computing Depth Without Prediction

A block diagram of the system used in this and succeeding sections is shown in Figure 3. The diagram represents two systems, one without prediction between frames and the other with prediction between frames (the multiframe algorithm). The point tracking module is common to both and is a correlation-based matching algorithm which operates on two successive frames: an image at time t_i and another at time t_s. A 3×3 window of data around each feature point is moved in discrete steps along the displacement path and correlation values between it and the corresponding windows in the second frame are obtained. The window location which results in the highest correlation is assumed to correspond to the displaced location of the feature point in the second frame. From the computed displacement, the depth of the point in the second frame is computed.

If we know the ground truth depth then we can compute the depth error at the point by computing the absolute difference between the computed depth and the ground truth depth. For this reason, the experiments described later in this section and in section 5 used a set of synthetic images generated by a ray tracing algorithm[Whit80]. Synthetic images were used, as opposed to an actual world

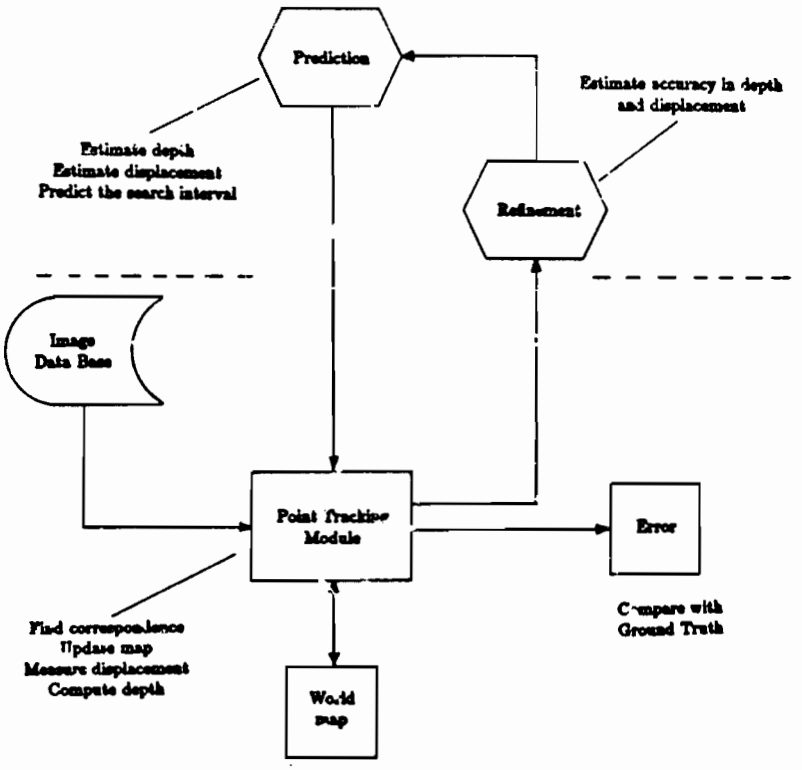


Figure 3: Block diagram of the multiframe algorithm.

scene, bec

- the images generated by the ray tracing algorithm have characteristics very much like those of real images and include shadows, reflections and transparency
- the camera geometry and the parameters for camera translation appropriate for our experiments were easily controlled during the image generation process;
- the ground truth depth map representing the actual depth of the points from the focal point of the camera can be conveniently obtained and compared with the depth map computed using the multiframe algorithm;

Figure 4 is a typical picture of a 256 × 256 image generated by the ray tracing algorithm. The camera is translated 0.2 cm. along the optical axis between any pair of consecutive frames. The optical axis of the camera is at $-15°$ relative to the positive $\hat{z}$-axis, and the translation of the camera causes points to be displaced by less than 7 pixels in the following image. Hence we use a search interval of 10 pixels (a conservative estimate) when searching for correspondence along the displacement path.

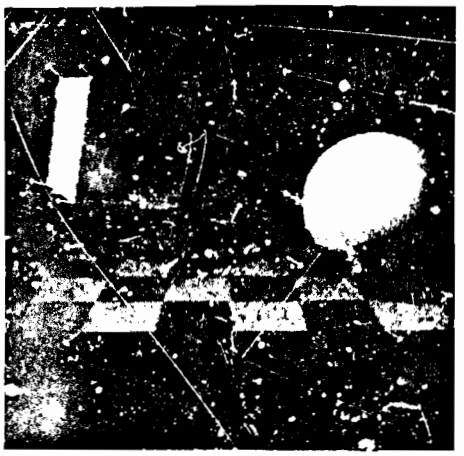


Figure 4: 256 × 256 image generated by the ray tracing algorithm. The vectors are unit vectors at each interest point along the displacement path.

4.1 Experiments

Several experiments using a sequence of five synthetic images labelled 0 through 4 have been performed to evalu-

ate the magnitude and source of errors in recovering depth maps over a sequence of image frames. In the first experiment depth is computed using overlapping pairs of consecutive frames without predicting feature displacements between frames. Hence, in the representation discussed in an earlier section, we only carry the x and y location of the points from one frame to the next. Therefore, each processing step of the algorithm consists of finding correspondences for each point, saving the x and y locations of each point, computing the depth, and computing the error in depth.

Results of the experiment at each processing step and for each feature point are shown in Table 1. The table indicates that the error in computed depth fluctuates from one step to the next. The trend is generally upward and accumulates to high values for a majority of the points by time t_4 when processing terminates. This is to be expected since errors in the location of the feature points have a tendency to accumulate.

In the second experiment we observe the depth computation from frames that are separated by larger camera translations. Results from this experiment are shown in Table 2. We find that for the same camera translations, the errors in the present case are smaller than those in the previous experiment.

As expected, the results from the second experiment

feature	true depth (cm.)	err. in comp. depth in (cm.)			
		0-1	1-2	2-3	3-4
1	13.505	0.501	0.899	1.299	1.698
2	13.132	4.459	4.879	5.271	5.665
3	11.970	1.701	1.507	1.313	1.118
4	11.833	1.753	1.428	4.171	0.776
5	11.833	1.511	1.319	1.127	0.934
6	9.902	4.559	4.721	4.860	5.693
7	9.902	2.480	9.127	3.327	10.365
8	9.902	2.990	3.341	0.798	4.044
9	8.951	3.017	2.606	2.198	11.534
10	8.951	2.951	2.529	2.110	8.951

Table 1: Results from the first experiment. In this experiment, only the positions of the feature points computed from a pair of frames (e.g. 0-1) are carried over to the next pair of frames (1-2). Errors in position are cumulative, resulting in large errors in the computed depth of the feature points, as shown in the last column.

show an improvement over those from the first one. There are two reasons for this improvement:

- the possibility of cumulative tracking errors was either eliminated (as in the case of 0-2,0-3 and 0-4) since only two frames were used or was minimized (as in the case of 0-2-4)[1];
- large camera translations give large displacements for points in the image and consequently lower relative errors in displacements.

feature	true depth (cm.)	err. in comp. depth in (cm.)			
		0-2	2-4	0-3	0-4
1	13.505	0.699	1.423	0.898	0.480
2	13.132	1.123	1.827	0.381	0.229
3	11.970	1.508	0.720	1.312	1.118
4	11.833	1.561	1.359	0.084	0.241
5	11.833	1.319	0.538	1.127	0.934
6	9.902	4.816	0.679	1.288	2.188
7	9.902	2.734	0.312	0.251	1.345
8	9.902	3.244	0.416	0.131	1.304
9	8.951	3.794	1.201	2.575	0.386
10	8.951	2.729	4.634	2.510	0.299

Table 2: Results from the second experiment. The table shows the results for three separate computations. The first two columns (0-2, 2-4) are a two step computation similar to that in Table 1, except that frames 1 and 3 were eliminated resulting in larger displacements of the feature points. The last two columns used only two pairs of frames each (0-3, 0-4).

These experiments identify some of the sources of errors which we had originally expected. First, the increase in the errors in computed depth is a result of the difficulty in successfully tracking points over a sequence of frames. Small errors in locating a match at each step are compounded over several processing steps. A consequence of matching errors is errors in displacements which in turn cause errors in computed depths.

Second, displacement can only be measured to within $\pm\frac{1}{2}$ of the resolution of the search for matching. Hence, the spatial resolution of the search limits the accuracy with which depth can be computed. Effects on accuracy due to noise are not applicable in our case because we use a controlled set of images [however, see Pavl85].

We believe that both the tracking and the matching problem can be addressed effectively by constraining the search interval (for matching) with depth information available from the preceding frames. In the next section we discuss an approach which requires only coarse estimates of depth from the early stages of processing and then refines it over a sequence of frames by searching at a finer resolution.

[1] 0-2-4 stands for an experiment in which depth was first computed by using frames 0 and 2 and then using the new locations of the points in frame 2, depth was computed from frames 2 and 4. This is different from 0-4 which represents depth computations directly from frames 0 and 4. Both the experiments compute the depth of the points in frame 4, yet the results from each could be different.

5 Prediction And Refinement

To recover depth maps of the environment reliably over a sequence of frames, it is important to reduce the possibility of matching errors. An improvement in matching is expected to positively affect the tracking problem. We have been using the following constraints to search for the displaced points:

- points can only move along the displacement path;
- points in the image can move no more than a fixed number of pixels between a pair of frames.

In addition, the following observation can be used to improve the matching:

- given a current estimate of the depth of a point and its accuracy, the displacement of the point in the next step can be predicted to fall within an interval along the displacement path.

In the start-up situation, the system has no prior depth information. This situation could arise either when a new set of images is input to the system or when the scene changes drastically, as in the case of turning around a corner. In this case, since the interval of search is large (because the accuracy is small), we minimise computation by searching at a coarse spatial resolution to obtain the depth estimates. Subsequent processing can use these estimates to reduce the search interval and conduct search at fine spatial resolution.

Once the depth map is determined accurately, any processing thereafter can be characterised by a similar prediction refinement cycle which can be aided by concurrent processing at lower temporal resolutions. By processing with frames separated by large time intervals, the depth computations at frame-rate (which defines the maximum temporal resolution) can be verified and corrected (if necessary).

5.1 Experiments

In this section, we describe preliminary experimental results from the multiframe algorithm. In this algorithm, the information carried forward from the analysis of frames t_i and t_{i+k} ($k \geq 1$) includes not only the predicted positions of the feature points but also the depth of the feature points as well as the expected accuracy of the depths. From the computed depths, a bound on the expected displacements of the feature points in frame t_{i+l} ($l > k$) is computed and used to limit the local search for the maximum of the correlation function.

An image sequence of 8 images labelled 0 through 7 was generated for this experiment. The camera translation between frames, the optical axis, and the axis of translation are identical to the ones specified for the previous experiment. To incorporate temporal resolution, we choose to use only frames 0,4,6 and 7, which represents an increase in temporal resolution by a factor of two at every step after the first one. Also, in order to improve the accuracy of depth computation by a factor of two, the spatial resolution of the search is 1 pixel in the first step and is increased thereafter by a factor of four for every subsequent step. The processing steps for the experiment can be described briefly as follows:

1. compute depth from image frames 0 and 4 by conducting a search at 1 pixel resolution to determine displacements;
2. use the depth estimates from step 1 to predict the displacements in frame 6;
3. repeat step 1 with frames 4 and 6, but conduct search at 1/4 pixel resolution;
4. use the depth estimates from step 3 to predict the displacements in frame 7;

feature	true depth (cm.)	expected accuracy in depth (± cm.)			err. in comp. depth (cm.)		
		0-4, 1 pix	4-6, $\frac{1}{4}$ pix	6-7, $\frac{1}{16}$ pix	0-4	4-6	6-7
1	12.923	0.924	0.360	0.159	0.816	0.049	0.278
2	13.908	1.776	0.639	0.333	1.098	0.327	0.307
3	13.510	1.423	0.533	0.231	1.522	0.358	0.123
4	12.387	2.617	1.866	0.676	1.118	1.780	0.243
5	12.256	1.304	0.710	0.294	1.025	0.235	0.388
6	10.239	0.770	0.293	0.128	1.346	0.722	0.460
7	10.239	1.047	0.378	0.160	1.034	0.212	0.121
8	10.327	3.744	1.107	0.436	5.450	2.727	1.777
9	9.366	1.169	0.399	0.164	0.785	0.100	0.444
10	8.350	0.436	0.162	0.075	0.301	0.038	0.109

Table 4: Results from the third experiment using prediction of the displacements of the feature points from frame to frame to constrain the local search. See text for details.

5. repeat step 1 with frames 6 and 7, but conduct the search at 1/16 pixel resolution

The results of the experiment are shown in table 3. A set of 10 interest points were selected for recovering a depth map over time. Values under the heading **Expected Accuracy In Depth** are those that were computed (using equation (3)) from the corresponding accuracy in measuring displacements at a given spatial resolution of search. Ideally, the error in computed depth should fall within the bounds of the expected accuracy in depth measurement at each step of the refinement process. We find this to be true for all the error values in step 1. At step 2 we find that the computed depth of 80% of the points is within the expected accuracy and this figure drops to 70% for step 3. Even though the actual error reduces from step 1 to step 3, the rate of reduction in error for some points after the first step is less than the rate of expected increase in accuracy (in depth) at higher spatial resolutions of search.

The observed discrepancies could result from the violation of the following assumptions which are implicit in our matching approach:

- matching is done around the global maximum of the correlation function;
- the correlation function decreases monotonically at points away from the global maximum;
- the global maximum can be measured with high accuracy;

If the global maximum of the correlation function cannot be measured with high accuracy, doing search at a high resolution will find only local peaks in the function and would not increase the accuracy in computed depth. Study of the shape of the correlation function in order to understand the limits on achievable accuracy is the subject of future work.

6 Future Work

In this paper we have identified several sources of errors in recovering environmental depth maps from known translational motion using multiple frames. We have also demonstrated through experiments, the role of prediction and refinement in dealing with the start-up problem and the problem of tracking points over a sequence of frames. These are preliminary studies only and much work remains to be done. The following areas in particular require more investigation.

1. The assumption that the global maximum of the correlation function has a sharp peak and decreases monotonically at points away from the peak is not valid at all points in an image. We intend to obtain a better characterisation of the shape of the correlation function in order to determine the accuracy with which the maximum of the function can be localised. Such an analysis will identify a spatial resolution of search which is best suited for the scale of the intensity variation at a point in the image.

2. We intend to explore the possibility of representing the image frames at multiple spatial resolutions in order to improve the matching scheme without sacrificing computational efficiency. In order to recognise low frequency variations in the image, it is necessary to use large correlation windows which tends to increase computation. By representing the image at a lower spatial resolution and using a smaller correlation window for matching[Glas83,Anan84], significant improvements in the computational effort can be realised. An implementation of the multiframe algorithm is possible which processes image frames at multiple spatial resolutions at each time step to obtain greater accuracy in matching and in tracking feature points over multiple frames.

3. We also intend to investigate the possibility of integrating depth information from processes at different temporal resolutions. As discussed earlier, there are definite advantages in projecting the depth estimates from processes at low temporal resolutions to those at high temporal resolutions.

4. We shall soon be evaluating the performance of the multiframe algorithm on sequences of images representing scenes from corridors and hallways inside a building as well as road scenes which are similar to those required for the ALV (Autonomous Land Vehicle) project.

References

[Adiv85] Adiv, G., Determining 3-D Motion and Structure from Optical Flow Generated by Several Moving Objects, *COINS Technical Report 84-07* University of Massachusetts, April 1984.

[Anan84] Anandan, P., A Confidence Measure for Correlation Matching, *Proc. of DARPA Image Understanding Workshop*, New Orleans, LA, 1984.

[Ball82] Ballard, D.H. and C.M. Brown, Computer Vision, Prentice Hall, Inc., 1982.

[Glas83] Glaser, F., G. Reynolds and P. Anandan, Scene Matching by Hierarchical Correlation, *IEEE CVPR conference*, June 1983, pp. 432-441.

[Kitc80] Kitchen, L. and A. Rosenfeld, Gray-Level Corner Detection, *TR-887*, Computer Science Center, Univ. of Maryland, April, 1980.

[Lawt84] Lawton, D.T., Processing Dynamic Image Sequence from a Moving Sensor, *Ph.D. Dissertation*

(TR 84-05) Computer and Information Science Dept. Univ. of Mass.(1984)

[Mora77] Moravec, H.P., Towards Automatic Visual Obstacle Avoidance, *Proc. of the 5th IJCAI*, MIT, Cambridge, MA, 1977.

[Tsai84] Tsai, R.Y. and T.S. Huang, Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces, *PAMI* 6 (1984), pp. 13-27.

[Ullm83] Ullman, S., Maximising Rigidity: The Incremental Recovery of 3-D Structure From Rigid and Rubbery Motion, *A.I. Memo No. 721*, MIT AI Lab, June, 1983

[Waxm83] Waxman, A.M. and S., Ullman, Surface Structure and 3-D Motion from Image Flow: A Kinematic Approach, *CAR-TR-24*, Center for Automation Research, Univ. of Maryland (1983)

[Whit80] Whitted, T., An Improved Illumination Model for Shaded Display, *Communications of the ACM*, 23(6), June 1980.

[Pavl85] Pavlin, L, Analysis of an Algorithm for Detection of Translational Motion, *in these proceedings.*

Acknowledgements

The authors want to thank P. Anandan for many helpful suggestions and comments throughout the course of this work. They also thank I. Pavlin for his image generator, and R. Heller, R. Melknap, B. Burns and M. Snyder for their help on several occasions.

GC970145-45

MULTIRESOLUTION PATH PLANNING FOR MOBILE ROBOTS

Subbarao Kambhampati
Larry S. Davis

Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742

ABSTRACT

The problem of automatic collision-free path planning is central to mobile robot applications. In this report, we present an approach to automatic path planning based on a quadtree representation. We introduce hierarchical path searching methods, which make use of this multiresolution representation, to speed up the path planning process considerably. Finally, we discuss the applicability of this approach to mobile robot path planning.

1. INTRODUCTION

The problem of automatic collision-free path planning is central to mobile robot applications. Path planning for mobile robots is in many ways different from the more familiar case of path planning for manipulators (see also the discussion in [Thorpe84]). For example:

(1) A mobile robot may have only an incomplete model of its environment, perhaps because it constructs this model using vision and thus cannot determine what is occluded by an object.

(2) A mobile robot will ordinarily negotiate any given path only once (as opposed to a manipulator which might perform the same task thousands of times). This implies that it is more important to develop a negotiable path quickly than it is to develop an "optimal" path, which is usually a costly operation.

(3) A mobile robot will be moving according to a previously computed path while it is computing an extension or modification to that path; also the path being developed may change as more information about the environment is discovered. Thus, path planning in mobile robots is a continuous online process rather than a single offline process as in the case of a manipulator.

Conventional path planning algorithms can be divided broadly into two categories. In the first category are the methods which make trivial (if any) changes to the representation of the image map before planning a path. The regular grid search [Thorpe84] and vertex graph methods [Moravec81] [Thompson77] [Nilsson69] fall into this category.

Though these methods keep the representational cost to a minimum, their applicability to mobile robot navigation is limited. For example, the regular grid search is [Wallace83] [Thorpe84] "too local" and its path planning cost increases with grid size rather than with the number of obstacles present. Further, both regular grid search and vertex graph methods generate paths which clip obstacle corners.

The methods in the second category make elaborate representation changes to convert to a representation which is easier to analyze before planning the path. Free space methods [Brooks83a], medial axis transform methods, Voronoi methods, etc., fall into this category. A potential practical shortcoming of such methods for mobile robot navigation is that the path planning cost is still very high because of the representation conversion process involved.

Though the above two categories by no means exhaust the existing methods (there are configuration space methods that use a vertex graph approach [Lozano79] and others that use a free space approach [Lozano81] to solve the manipulator findpath problem), they do point out that what mobile robots need may be a compromise between these two categories.

It is these considerations that motivated the multiresolution (hierarchical) representation based path planning algorithms described in this report (see also [Davis85]). Similar considerations also led to the use of hierarchical representations in manipulator "findpath" problems (see Section 4 for a discussion of related work). In this report, we first develop a method of path planning for mobile robots using a hierarchical representation based on quadtrees and then describe staged search as a way of exploiting the hierarchical nature of the representation to gain substantial computational savings. Throughout this report we restrict our attention to two-

The support of the Defense Advanced Research Projects Agency and U.S. Army Night Vision and Electro Optics Laboratory under Contract DAAK70-83-K-0018 is gratefully acknowledged

dimensional path planning without rotation and a vehicle with circular cross-section.

Section 2 develops a quadtree planning algorithm based on A^* search. Section 3 presents a staged (hierarchical) path planning algorithm which has computational advantages as compared to the pure A^* search on quadtrees. The staged search involves inclusion of gray nodes in the search. Section 4 discusses related work and finally, Section 5 summarizes the conclusions reached from this research and discusses future directions. In the remainder of this section we define some terms used in these discussions.

Quadtree related terminology: A *quadtree* is a recursive decomposition of a 2-D picture into uniformly colored $2^i \times 2^i$ blocks (e.g., see Figure 2.1) [Samet83]. A node of a quadtree represents a $2^i \times 2^i$ square region of the picture. A *free node* of a quadtree is a node of the quadtree representing a region of freespace. An *obstacle node* is a node representing a region of obstacles. A *gray node* is a node representing a region having a mixture of freespace and obstacles. A *leaf node* of a quadtree is a tip node of the tree. In ordinary quadtrees, leaf nodes are always obstacle nodes or free nodes, but in pruned quadtrees (see below), they may also be gray nodes. For any gray node G, $S(G)$ denotes the subtree rooted at G. $L(G)$ denotes the number of leaf nodes in $S(G)$. The *gray content* of a gray node G is defined as the number of obstacle pixels in the region represented by G, and the *grayness* of G is the percentage of obstacle pixels in that region.

A pruned quadtree, Q_p, of a quadtree, Q, is generated by making some of the gray nodes, G_i, into leaf nodes, thus pruning the subtrees, $S(G_i)$, rooted at the G_i. The pruned quadtree, Q_p, thus represents the same space as Q, but with a reduced resolution.

A^* terminology: A^* is a classical minimum cost graph search algorithm, whose optimality properties are well known [Nilsson80]. In this algorithm OPEN is a list consisting of all the nodes in the search graph that are generated but not yet expanded. CLOSED is the list of nodes in the graph that have been expanded. *Best node* is the node that is currently being expanded in the search. This node has the best evaluation (i.e., minimal path cost) among the nodes on OPEN. The *predecessor* of a node N in the search graph is the node preceding N on the current best path to N (from the start node).

2. QUADTREE BASED PATH PLANNING

2.1. Representation preprocessing

We have developed an algorithm for mobile robot path planning based on a quadtree representation of the robot's immediate environment. If there are large areas of free space (or of obstacles) then those areas can be represented by a few large blocks in the quadtree and can be dealt with as units by the planning algorithm.

Given a binary array or raster representation of a robot's immediate environment we first grow the obstacles by the radius of robot's cross-section [Moravec81] and then convert the raster into a quadtree representation using a raster to quadtree conversion algorithm [Samet81]. This algorithm is of complexity $O(n)$ where n is the number of pixels in the image being converted. In the resulting quadtree blocks of 0's represent free space nodes and blocks of 1's represent obstacle nodes.

In the second stage of preprocessing, we compute the *distance transform* of the set of 0's, i.e., the free space blocks. This determines, for each block of free space, the minimal distance between the center of that block and the boundary of a block of obstacles. Samet [Samet82a] describes an algorithm for computing this distance transform for quadtrees which is of complexity $O(n)$, where n is now the number of leaf nodes in the quadtree.

2.2. Path planning algorithm

Given the start and goal points, we first determine the quadtree leaf nodes, S and G, representing the regions of the image containing these points. Next, we plan a minimum cost path between S and G in the graph formed by the non-obstacle leaf nodes of the quadtree, using the well known A^* search algorithm with the evaluation function f of a node c defined as

$$f(c) = g(c) + h(c)$$

Here $g(c)$ represents the cost of the path from S to c and $h(c)$ represents the heuristic estimate of the cost of the remaining path from c to G.

Since the cost of a path should depend both on the actual distance travelled along the path and the clearance of the path from the obstacles, we define $g(c)$ as

$$g(c) = g(p) + \hat{g}(p, c)$$

where $g(p)$ is the cost of the path from S to c's predecessor, p, on the path and $\hat{g}(p, c)$ is the cost of the path segment between p and c. The latter function in turn is defined as

$$\hat{g}(p, c) = D(p, c) + \alpha\, d(c)$$

with $D(p,c)$ representing the actual distance between nodes p and c, given as half the sum of the node sizes, and $d(c)$ representing the cost incurred by including node c on the path. $d(c)$ depends upon the clearance of the node c from the nearby obstacles. We chose a linear shape for the cost function d, defining $d(c)$ as

$$d(c) = o_{max} - o(c)$$

where $o(c)$ is the distance of the node c from the nearest obstacle given by the quadtree distance transform, and o_{max} is the maximum such distance for any node in the quadtree (so that $d(c)$ is always positive). α in the equation for $g(p, c)$ is a positive constant which determines by how far the resultant path will avoid obstacles.

Finally, $h(c)$ is calculated as the Euclidean distance between the midpoints of the regions represented by c and G. Clearly, this measure is a lower bound on the actual minimum cost path between c and G; thus an A^* search with this measure as its heuristic estimate is admissible. The power of this heuristic depends upon the average deviation of the minimum cost path from the straight line path. It is highest for the case where α is zero and decreases as α increases. It is of course possible to use more informed, but inadmissible, heuristics to speed up this search.

The node expansion process involves finding the non-obstacle leaf nodes adjacent to the the node being expanded. We accomplish this by using a neighbor finding strategy similar to that given by Samet [Samet82b] with two differences. First, only the neighbors in the horizontal and vertical directions are considered—diagonal neighbors, which share only single points with the current node, would result in inflexible paths which clip obstacle corners. Secondly, when one of the neighbors given by the quadtree neighbor finding algorithm is a gray node, we find the non-obstacle leaf nodes, if any, of the quadtree rooted at that gray node that are adjacent to the node being expanded and consider them as neighbors.

The result of applying the above A^* algorithm to the quadtree is a list of nodes from the quadtree (ordinarily of varying sizes) which define a set of paths between the start and goal nodes. If desired, an optimal path through these blocks can be computed, or the center points of consecutive blocks on the list can be connected to compute a negotiable path.

2.3. Results

Figure 2.2 contains a simple example of a path obtained using this algorithm. Figure 2.2(a) is a binary array with start and goal points marked, along with an indication of the path determined by the algorithm. Figure 2.2(b) contains the tree data structure that represents the quadtree, in which the blocks on the computed path are marked with p's. It is important to note the reduction in the number of nodes achieved by the algorithm. Figure 2.3(a) shows a path planned on a more complicated image map with the constant α set to 1 and Figure 2.3(b) shows the same example with α set to zero. Notice that the time taken in the former case is considerably higher than in the latter. This should be expected, since as noted in the last section, the heuristic power of h reduces as α increases.

It is also interesting to note that although it is true that the quadtree representation is sensitive to displacements of obstacles with respect to the grid boundaries, the savings in space and computation afforded by this method are still very high on the average. Further, Samet et al. [Samet83] point out that for complicated images the positioning of the image origin is likely to have little effect on the number of nodes in the resultant quadtree.

2.4. Advantages of the quadtree approach

Compared to the first category of path planning algorithms mentioned in the introduction, such as the grid search method, the path planning cost for quadtree based search will be substantially lower because the number of nodes to be searched in the quadtree approach is considerably smaller. In fact, the number of leaf nodes in a quadtree of an image map having polygonal obstacles is approximately [Samet83] $\frac{2}{3}O(p)$, where p is the sum of the perimeters of the (polygonal) obstacles in terms of the lowest resolution units, in our case pixels (or grid points). Thus A^* search will only have to deal with about $O(p)$ nodes in the case of a quadtree, instead of the n^2 grid points in the case of a grid search, a substantial reduction. Similarly the "local-bound" behavior of the first category algorithms is absent in this approach, because the nodes are on the average much larger than single pixels and it is straightforward to determine the "nearness" of the nodes to the obstacles. A hierarchy of different levels of description of the space that is available with quadtrees enables us to search for a path close to obstacles only when necessary. Corner-clipping, inflexible paths are eliminated by considering only neighbors in the horizontal and vertical directions.

Unlike the second category of methods that involve a costly change of representation, the proposed approach has a very small representation overhead. As pointed out in Section 2.1, both the representation algorithms involved are of complexity $O(n)$, whereas many methods of the second category have a representation cost which is far higher.

Thus quadtree based path planning is a good compromise between free space algorithms and grid search type algorithms. In addition, the path produced by the quadtree algorithm, although not "optimal", is a "negotiable" path which can be computed relatively quickly. Apart from this, the hierarchical nature of the representation gives many advantages in path planning. For example,

[a] We can easily constrain the path to satisfy certain conditions such as specification of minimal clearance of the path.

[b] More importantly, we can make the search *staged*, i.e., plan a path at a coarser level and subsequently refine it as needed, thus reducing the planning cost substantially.

The former advantage has been discussed in Section 2.2. We will discuss the latter at greater length in the next section.

3. STAGED PATH PLANNING

3.1. Motivation

Though the algorithm which we presented in the previous section is relatively efficient it can be improved upon substantially. We often get undesirably small "black" (obstacle) nodes in the quadtree representation. One obvious source for this may be the existence of very

small obstacles in a region of the environment that is otherwise obstacle free. A more important source of these black nodes is the representation of irregular obstacles in quadtrees. Due to the recursive nature of the quadtree, these small black nodes will fragment the free space, giving rise to an undesirable increase in the depth of the quadtree and the number of leaf nodes, consequently increasing the cost of the search.

We can deal with this problem by first planning the path in a reduced resolution quadtree, called a *pruned quadtree*, that contains gray leaf nodes, corresponding to mixtures of obstacles and free space. This implies that a node can now have gray neighbors. An algorithm which is capable of planning a global path at this coarser level, and subsequently developing the path inside the gray nodes (which are included in the global path) in the second stage, can give rise to savings in terms of computation, without significant degradation of the path obtained. As mentioned in Section 2.4, the number of leaf nodes is on the order of the sum of the perimeters of the obstacles, measured in the lowest resolution units. Thus conducting search at a resolution l levels below the pixel resolution reduces the "sum of the perimeters" and "number of leaf nodes" by a factor of 2^l thereby substantially reducing the time complexity of the search.

There are two aspects to this staged search that deserve detailed attention—the treatment of gray leaf nodes during planning and the generation of pruned quadtree from the original quadtree. In the next two subsections we shall discuss these two aspects in detail.

3.2. Dealing with gray leaf nodes

When planning a path through the pruned quadtree, we have to deal with gray leaf nodes. Specifically, the following three questions must be answered:

(1) What is done when one of the neighbors of the current best node (the node that is currently being expanded in the A^* search) is a gray node?

(2) How is the current path expanded when the current best node is a gray node?

(3) How is the first stage path, involving gray leaf nodes, processed to get the final path that contains free nodes exclusively?

We shall address these in the following subsections.

3.2.1. Gray leaf neighbors

If one of the neighbors, N, of the current best node, B, is a gray node then before putting N on the OPEN list, we must ensure that N can be entered from B. If B is a free node then N can be entered iff there exists at least one free node, m, in $S(N)$ such that m is adjacent to B. If, in addition, B itself is a gray node then N can be entered from B as long as there exists a free node, e, in $S(B)$ such that e is adjacent to m. Note that checking this entry condition alone does not guarantee that the gray node N is passable, i.e., that a path from B through N to a third node, C, exists. For example in Figure 3.1, N can be entered from B, through the free node m, but N cannot be exited, except back to B.

If we decide to put N on the OPEN list then we shall include in the heuristic value of N a measure of the "path complexity", $c(N)$, inside N. (This measure should be zero for a free node since the path inside the free node can be a straight line.) In general, it is difficult to give a measure which truly represents the complexity of a path inside the gray node, since at this point in the search the direction in which the path will be exiting the gray node is unknown. But in practice any measure depending upon the gray content (number of obstacle pixels inside the gray node) of the gray node will be a good choice. One such normalized complexity measure for the gray node N is

$$c(N) = \frac{gray\ content(N)}{size(N)}$$

Given two gray nodes having the same gray content, the path complexity should intuitively be higher for the gray node representing a region with more obstacle nodes. Thus, a better, although costlier, complexity measure of the gray node N will take into account the number of obstacle nodes in $S(N)$.

Once the heuristic value is calculated, the gray node is placed on the OPEN list and it can be selected for expansion whenever its f-value is the best among the nodes on the OPEN list.

3.2.2. Expanding gray nodes during search

When the current best node, B, happens to be a gray node, expanding B becomes a more involved operation. After generating B's neighbors we must ensure that for each of these neighbors, N, there exists a path through B that connects B's predecessor, P, on the current path to N (see Figure 3.2). We refer to this as the "reachability" analysis for neighbor N. Secondly, for each neighbor N that can thus be reached we have to record as N's g-value an estimate of the shortest path to N through B. This estimate should take into account the fact that the shortest path through B may not be a straight line path since B is a gray node.

One way to achieve the above two objectives is by performing an A^* search rooted at B to determine if N can be reached from P. If the A^* search finds such a path to N, then we can use the cost of that path as the g-value of neighbor N. The advantage of this method is that we have the full power of A^* search. The principal disadvantage to this method is that we need to perform this A^* search once for every neighbor of B, a rather large price to pay for path optimality.

To avoid the above disadvantages associated with A^* search we elected to compute a distance transform of the gray node as a way of dealing with the problems of best gray node expansion.

Let f be a free node in $S(B)$ such that f is adjacent to B's predecessor, P. Notice that there can be more than one such free node in $S(B)$. If P is a gray node, then we require that f be adjacent to a free node in $S(P)$ (called an "exit node" for P). This exit node would have

been determined when P was being expanded. We illustrate all this in Figure 3.2. P is the predecessor of the best node, B, and N is a neighbor of B. Both f and f' are free nodes in $S(B)$. They are also adjacent to P. In such a situation, we chose the free node which has the least straight line distance to the goal node—in this case f. Thus the current path enters B through f. f is recorded as the *entry node* of B.

Next, we compute a distance transform of the region represented by B, with respect to f. This involves recording for each free node, f', in $S(B)$, f''s shortest distance (which we refer to as $dis(f, f')$) from f. To carry out this computation, we first initialize $dis(f, f)$ to zero (see Figure 3.2), and $dis(f, f')$ for all other free nodes, f', in $S(B)$ to ∞. Next, we carry out the propagation step: we find all the neighbors of f, f', which are in $S(B)$ and for each such neighbor, f', calculate $dis(f, f')$, as the sum of $dis(f, f)$ and the nodal distance between f and f', $D(f, f')$. To ensure that the path inside B will take clearance from the obstacles into consideration, we include the cost of the node $d(f')$ (see Section 2.2) in $dis(f, f')$. We repeat this propagation step for all the neighbors of f, with the neighbors taking the role of f, and so on, until we exhaust all the free nodes in $S(B)$. The detailed procedure is given in an algorithmic fashion in Table 3.1. and is, essentially, the familiar shortest path algorithm for the case of "single source multiple destinations" (see, e.g., [Horo82]).

Having computed the distance transform of B with respect to f, as detailed above, we are now ready to continue with the expansion of B. For each of B's neighbors, N, N is marked reachable, if there exists a free node, e, in $S(B)$ which satisfies the following two conditions (see Figure 3.2.):

(1) $dis(f, e) < \infty$. This ensures that there is a path between e and f inside $S(B)$.

(2) N can be entered from e. As discussed in Section 3.2.1, if N is a free node, this condition is satisfied as long as N and e are adjacent. If, on the other hand, N is a gray node then the condition is satisfied if there exists a free node m in $S(N)$ such that m and e are adjacent.

The node, e, satisfying the above two conditions is marked as the exit node of gray node B with respect to N. Notice again that there may be more than one such node. For example, in Figure 3.2, e and e' satisfy both conditions, since there is a path from f to each of these nodes, and N can be entered from both the nodes. In such a situation, we select the node with smaller distance to f as the exit node. Thus, in Figure 3.2, e would be chosen as the exit node of B with respect to N.

Neighbor N of the best node B, is placed on the OPEN list only if there exists an exit node, e, for B with respect to N. If N does go on to the OPEN list, the sum of the g-value of B's predecessor P, $g(P)$, and $dis(f, e)$ is recorded as $g(N)$. If N is a gray node, we have to include in N's heuristic value, $h(N)$, an estimate of the path complexity inside N as discussed in Section 3.2.1. This completes the discussion of the expansion of the best gray node B.

At this point it is worth noting the advantages of using the distance transform in dealing with gray leaf nodes: First, it eliminates the necessity of multiple rooted A^* searches. The distance transform computation is efficient on the quadtree representation. Second, developing the path inside the gray nodes, after the first stage, is very simple (see below)

3.2.3. Developing the first stage path containing gray nodes

At the end of the first stage of the staged search the planned path may contain gray nodes as well as free nodes. The path inside the gray nodes is developed in the second stage.

If rooted A^* search were used in expanding gray nodes (as discussed in the previous subsection), then this second stage would simply amount to concatenating these paths through gray nodes with the free nodes.

If the distance transform is used instead of rooted A^* search, then the path development inside gray nodes is not as simple. The path development computation involves the following (refer again to Figure 3.2):

For each gray node B on the path we retrieve B's entry node f (recorded while expanding B) and B's predecessor P and successor N on the path. Next, using N, we retrieve the exit node, e, for B corresponding to N. Now developing the path inside B amounts to finding the shortest path between e and f and inserting it in between N and P. Finding the shortest path between e and f simply involves backing up to f through neighbors having smallest distance transform values. In Figure 3.2, for example, the shortest path between e and f, as found by this method, is $e - d_1 - d_2 - \cdots - d_n - f$.

3.3. Pruned quadtree generation methods

The primary motivation for pruned quadtree based staged search, as noted in Section 3.1, is to offset the disadvantages of the fixed grid uniform recursive decomposition involved in quadtree representation. By choosing an appropriate pruned quadtree, we can avoid a profusion of nodes in a region of the image map which is relatively obstacle free. This poses the question of how to decide when a region, or the gray node representing it, is relatively obstacle free. None of the simple measures (such as grayness of the node) alone can answer this question entirely satisfactorily. For example, the grayness of a node tells us nothing about the distribution of the obstacles in the region represented by that node, and in the extreme case a small value of grayness may actually be the result of a streak of obstacle pixels through the middle of the node. More commonly, a small grayness value of a gray node may be due to a scattered obstacle distribution inside the gray node, which fragments the free space. In such a case, the gray node is obviously a bad candidate for a leaf node in the pruned quadtree. At the same time, we do not want to base our decision on a very involved analysis of the gray node, because this may

increase the cost of pruned quadtree generation to the point where the staged search is, overall, less efficient than searching the original quadtree.

The method proposed uses a threshold on $L(G)$, the number of leaf nodes in $S(G)$ to identify leaf nodes of the pruned quadtree. Any gray node, G, whose $L(G)$ is lower than the threshold is made a leaf node of the pruned quadtree in a breadth first traversal of the quadtree. Computation of $L(G)$ is straightforward. For a given threshold, there is an upper bound on the cost of gray node evaluation based on the distance transform, and thus the cost of the staged search can be effectively controlled.

One important advantage of this method is that the threshold on $L(G)$ is relatively independent of the specific image, and depends only on global criteria such as maximum allowable gray node evaluation cost and maximum allowable suboptimality of the resultant path. Figure 3.3(b) shows a pruned quadtree generated using this method from the quadtree in Figure 3.3(a) and also gives the result of a staged search on this pruned quadtree.

3.4. Results of the staged search

Figures 3.4(I-IV) depict the paths found by pure A^* search on the original quadtree, the first stage of staged search on the pruned quadtree (with gray nodes in the path), and the second stage of staged search (after paths inside the gray nodes are developed). The pruned quadtree used in the staged search is generated automatically, as discussed in the section on pruned quadtree generation. Each of the figures lists the cpu time taken for path planning, number of nodes considered by the search versus total number of leaf nodes, and details of the method of pruned quadtree generation used.

The path generated by the staged search is comparable to the optimal path generated by the pure A^* search. However, the total cpu time taken (with compiled Franz Lisp running on a VAX11/785) by staged search (for pruned quadtree generation, A^* search and second stage path development) is 3 to 10 times less than that taken by the pure A^* search. (See Figures 3.4(I-IV) for detailed timings for the examples presented. The timings are in cpu seconds and involve substantial page swapping overhead.) Our experiments show that the computational savings are much higher for cluttered environments than for relatively free environments—compare Figures 3.4(I) and 3.4(IV), for example. This is reasonable since the fragmentation of free space is much higher in cluttered environments.

4. RELATED WORK

As pointed out in Section 1, hierarchical representations have been used previously in manipulator findpath tasks. In this section we discuss some of that previous work in relation to our own.

Wong et al. [Wong85] use a modified version of quadtrees to solve 3-D findpath problems by planning a path in the three orthogonal 2-D projections of the 3-D environment. Their approach essentially searches for a path in a "point based" quadtree representation. (See [Samet83] for a comparison between "region based" and "point based" quadtrees.) Faverjon [Faverjon84] uses octrees (an extension of quadtrees to 3-D) for reducing the time complexity of the 3-D findpath problem for a six joint manipulator.

Lozano-Pérez [Lozano81] represented free space in the "configuration space" as a hybrid hierarchical structure consisting of rectanguloid and polyhedral cells. He, however, planned a cell path strictly among the free cells of the representation, thus missing the computational advantages of hierarchical staged search. Another problem with his approach was that the path search could fail because the resolution of the representation was not fine enough. Brooks and Lozano-Pérez later remedied these problems in [Brooks83b]. The approach presented in their paper comes closest to our "staged search" approach. They cut the free space hierarchically into full (obstacle), empty (free), and mixed rectanguloid cells, with the mixed cells representing areas of unexplored configuration space. They first try to plan a path exclusively through the free cells. If that fails, they then repeat the search, this time considering the mixed cells also. Next, for each mixed cell in the cell path, they try to develop a path through the mixed cell. If any of the mixed cells turns out to be impassable, then they may have to repeat the search again, finding another free-mixed cell path. Since they use the A^* search algorithm as the main engine for all these different searches, the overall process turns out to be very expensive computationally. Both [Lozano81] and [Brooks83] refine their cell paths into point paths, since the cell path in configuration space represents a set of possible solutions to the findpath problem.

5. CONCLUSIONS

In this report we have presented methods of short range path planning for mobile robots, using quadtree hierarchical data structures. We demonstrated the merits of quadtree based path planning and also discussed in detail a method of staged path planning, with improved computational cost compared to pure quadtree based single stage path planning.

Lozano-Pérez [Lozano81] observes that the most important heuristic for a path planning space representation is to avoid excess detail (and therefore time spent) on parts of the space which do not affect the planning operation. The quadtree representation naturally provides such a description of free space. Short range planning for a mobile robot should be based on decomposition of free space into units larger than pixels for the search to be global. Hierarchical decompositions like the quadtree are a good way to achieve this, especially since the representation cost involved is small. They obviously are not as optimal as decomposing free space into channels or more natural shapes, but the latter methods have a higher representation cost. Some of the suboptimality of

uniform grid recursive decomposition involved in quadtree representation is offset by the staged version of the path planner. Another important use of staged search in dynamic path planning is that it offers an elegant way of treating uncharted areas. These can be represented as gray nodes with very high co-, and when they get included in the search, further processing can be expended to "chart" those regions.

Looking further, the mobile robot needs to continually update the planned path, as it traverses it, in the light of new information. To do this efficiently, we need to be able to "add to" and "delete from" (or update) the representation of the image map with relatively low cost. In the context of dynamic path updating, one desirable property of a free space representation is that the individual obstacles affect the representation only in their immediate locality. A disadvantage of quadtree representation of free space is that it does not localize the effect of obstacles on the representation. This is a general shortcoming of representations which cut free space into rectanguloid cells. In contrast, the generalized cone representation of free space described in [Brooks83a] satisfies this property. Presently we are concentrating on efficient methods of path updating for the quadtree based planning methods discussed in this report.

REFERENCES

[Brooks83a]
Brooks, R.A., "Solving the findpath problem by good representation of free space," *IEEE Transactions on Systems, Man, and Cybernetics* **13**, 1983, 190-197.

[Brooks83b]
Brooks, R.A. and Lozano-Perez, T., "A subdivision algorithm in configuration space for findpath with rotation," in *Proceedings, Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, W. Germany, 1983.

[Davis85]
Davis, L.S., Andresen, F., Eastman, R., Kambhampati, S., "Visual algorithms for autonomous navigation," in *Proceedings IEEE International Conference on Robotics and Automation*, St. Louis, MO, March 1985.

[Faverjon84]
Faverjon, B. "Obstacle avoidance using an octree in the configuration space of a manipulator," in *Proceedings, IEEE International Conference on Robotics*, Atlanta, GA, March 1984.

[Horo82]
Horowitz, E. and Sahni, S., *Fundamentals of Data Structures*, Chapter 6, Computer Science Press, Rockville, MD, 1982.

[Lozano79]
Lozano-Perez, T. and Wesley, M.A., "An algorithm for planning collision-free paths among polyhedral obstacles", *Communications of the ACM* **22**, 1979, 560-570.

[Lozano81]
Lozano-Pérez, T., "Automatic planning of manipulator transfer movements," *IEEE Transactions on Systems, Man, and Cybernetics* **11**, 1981, 681-698.

[Moravec81]
Moravec, H., "Rover visual obstacle avoidance", in *Proceedings, Seventh International Joint Conference on Artificial Intelligence*, Vancouver, B.C., Canada, 1981.

[Nilsson69]
Nilsson, N.J., "A mobile automaton: an application of artificial intelligence techniques," in *Proceedings, First International Joint Conference on Artificial Intelligence*, Washington, DC, 1969.

[Nilsson80]
Nilsson, N.J., *Principles of Artificial Intelligence*, Chapter 2, Tioga, Palo Alto, CA, 1980.

[Samet81]
Samet, H., "An algorithm for converting rasters to quadtrees," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **3**, 1981, 93-95.

[Samet82a]
Samet, H., "Distance transform of images represented by quadtrees," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **4**, 1982, 298-303.

[Samet82b]
Samet, H., "Neighbor finding techniques for images represented by quadtrees," *Computer Graphics and Image Processing* **18**, 1982, 37-57.

[Samet83]
Samet, H., "The quadtree and related hierarchical data structures," University of Maryland Center for Automation Research Technical Report 23, November 1983.

[Samet84]
Samet, H. et al, "Application of hierarchical data structures to geographical information systems - Phase III," University of Maryland Center for Automation Research Technical Report 99, p. 59, November 1984.

[Thompson77]
Thompson, Alan M., "The navigation system of the JPL robot," in *Proceedings, Fifth International Joint Conference on Artificial Intelligence*, Cambridge, MA, 1977.

[Thorpe84]
Thorpe, C., "Path relaxation: path planning for a mobile robot," in *Proceedings, National Conference on Artificial Intelligence*, Austin, TX, 1984.

[Wallace83]
Wallace, R., "Two-dimensional path planning and collision avoidance for three-dimensional robot manipulators," in *Representation and Processing of Spatial Knowledge*, University of Maryland Department of Computer Science Technical Report 1275, May 1983.

[Wong85]
Wong, E.K. and Fu, K.S., "A hierarchical orthogonal space approach to collision-free path planning," in *Proceedings, IEEE International Conference on Robotics*, St. Louis, March 1985.

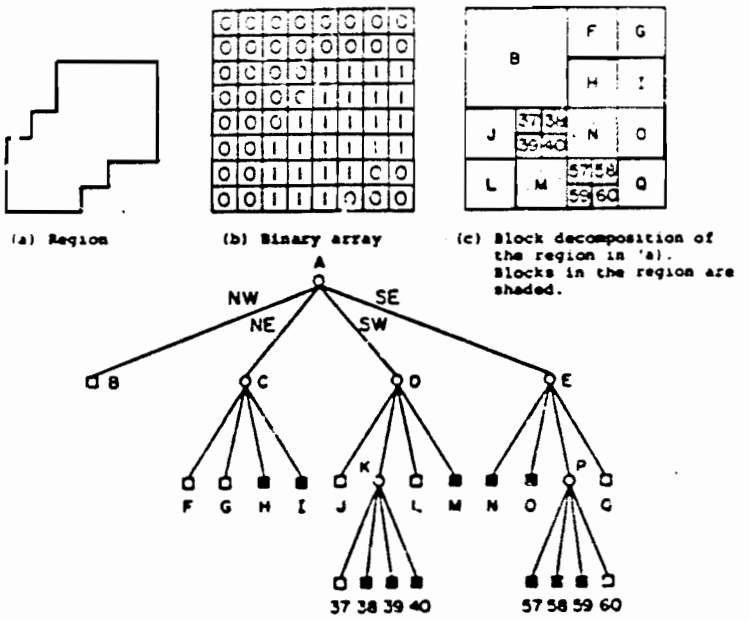


Figure 2.1

A region, its binary array, its maximal blocks and the corresponding quadtree (from [Samet83,])

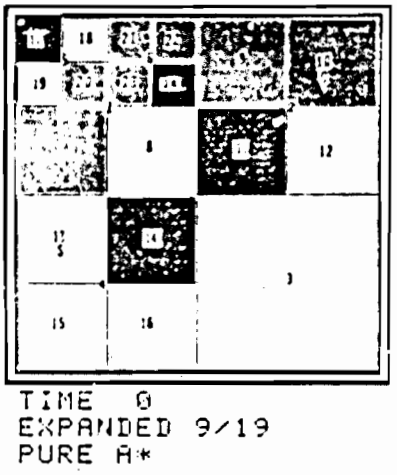


Figure 2.2(a) Path planned on a quadtree representation

A binary image with obstacles represented by black regions. Start node is indicated by S and goal node is indicated by G. The nodes on the path found by the algorithm are represented by shaded regions. All the node boundaries are outlined with black lines.

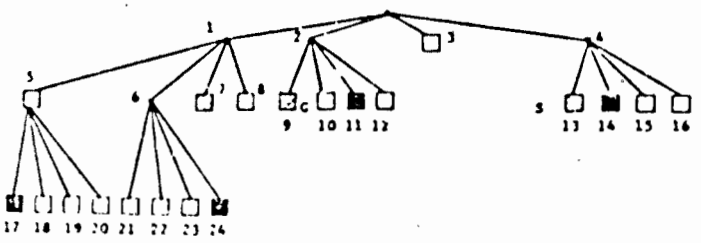
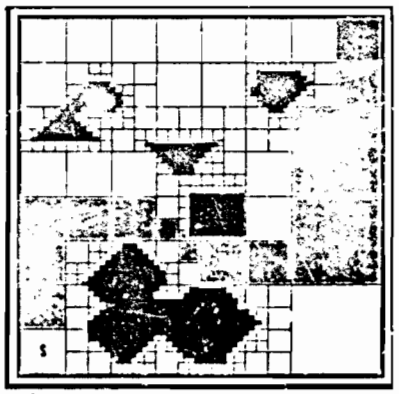


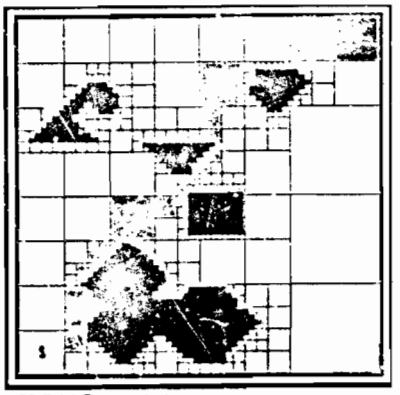
Figure 2.2(b) A tree data structure representing the quadtree of the binary image in Figure 2.2(a)

The black nodes correspond to the obstacle regions. The nodes corresponding to start and goal points are marked by S and G. Shaded nodes represent the nodes that fall on the path generated by the algorithm.



(a)

Result of single stage A^* search on the pure quadtree representation of an image, with α set to 1. S and G represent start and goal nodes and shaded regions represent nodes on the planned path



(b)

The same example as in (a), with α set to 0, resulting that the planned path need only barely avoid obstacles

Figure 2.3 Example of single stage planning

Procedure Distrans(B, f);

• B is the gray node representing the region in which f is a free node. The algorithm computes the distance transform of B with respect to f •

begin
  $\forall_{node} \in B$ $dis(f, node) \leftarrow \infty$;
  $dis(f, f) \leftarrow 0$;
  $distrans_OPEN \leftarrow$ bold list (f);
  until null $(distrans_OPEN)$
    do
      $f \leftarrow$ **first** $(distrans_OPEN)$;
      $distrans_OPEN \leftarrow$ **rest** $(distrans_OPEN)$;
      $NBRS \leftarrow$ get_neighbors_inside_the_node(f, B);
      foreach $nbr \in NBRS$
        do
          $dis(f, nbr) \leftarrow$ **min**{D (f , f) + α node_cost(nbr)
                + $dis(f, f)$,
              $dis(f, nbr)$}.
          $distrans_OPEN \leftarrow$ **append** $(distrans_OPEN, nbr)$;
        od
  od
end Distrans.

Table 3.1 Distance transform algorithm

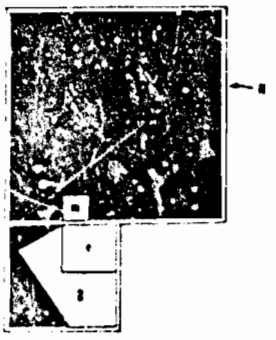


Figure 3.1 Dealing with a gray neighbor

The gray neighbor N, is placed on OPEN since there is a free node, m, in N's cell that is adjacent to the free node, n, of the best node, B, corresponding to N. Note that this does not guarantee that N is passable since N cannot be routed to any node other than B.

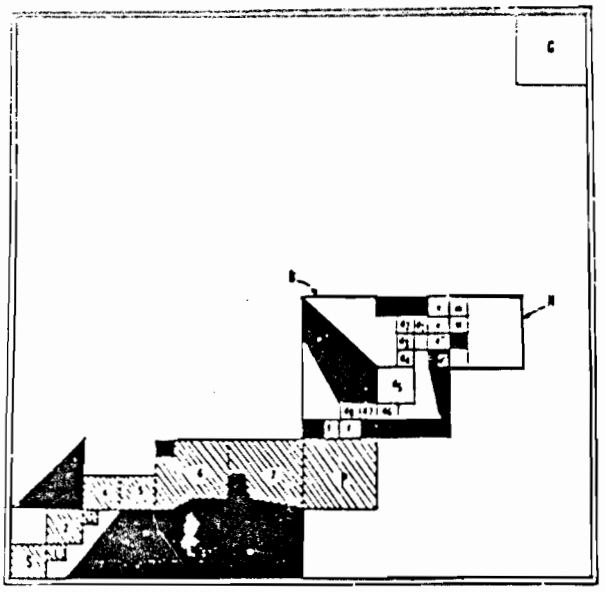
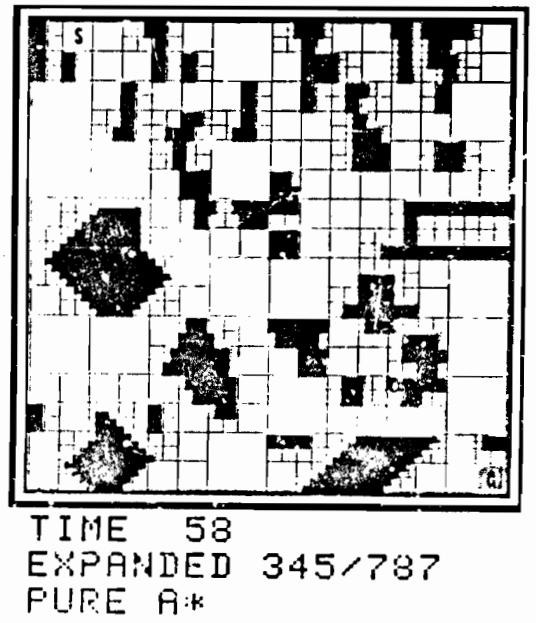


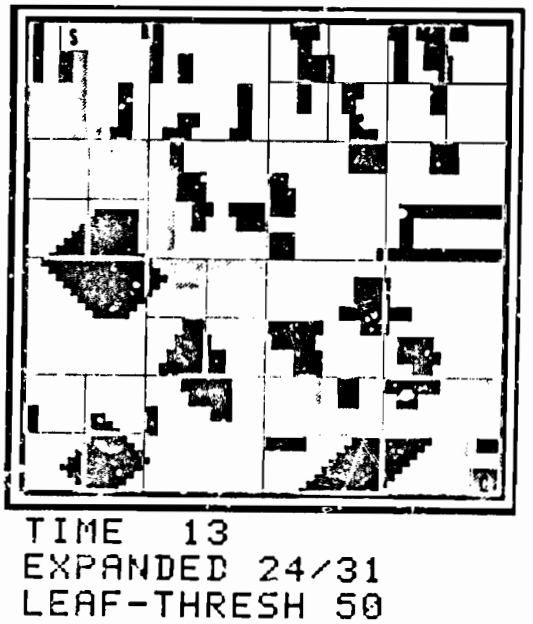
Figure 3.2 Expanding a gray node

In the figure, B represents the current best node in the A^* search and N is one of its neighbors. S and G represent start and goal nodes respectively. The path to B consists of nodes $S-1-2-3-4-5-6-7-P-B$ in that order. Thus P is B's predecessor on the current path. Of the two nodes f and f' that are adjacent to P, f is nearer to G, so f is the entry node of B. Of the four nodes e, e', e'', and e''' that are adjacent to node N, N can not be entered from e'' and e''' can not be reached from f. Thus e, e' are the possible candidates for exit nodes. e is chosen as the exit node for B corresponding to N since it is nearer to f than e'. $f-d_1-d_2-\cdots-d_n-f$ represents the path that will be developed inside B during the second stage of path development.



(a)

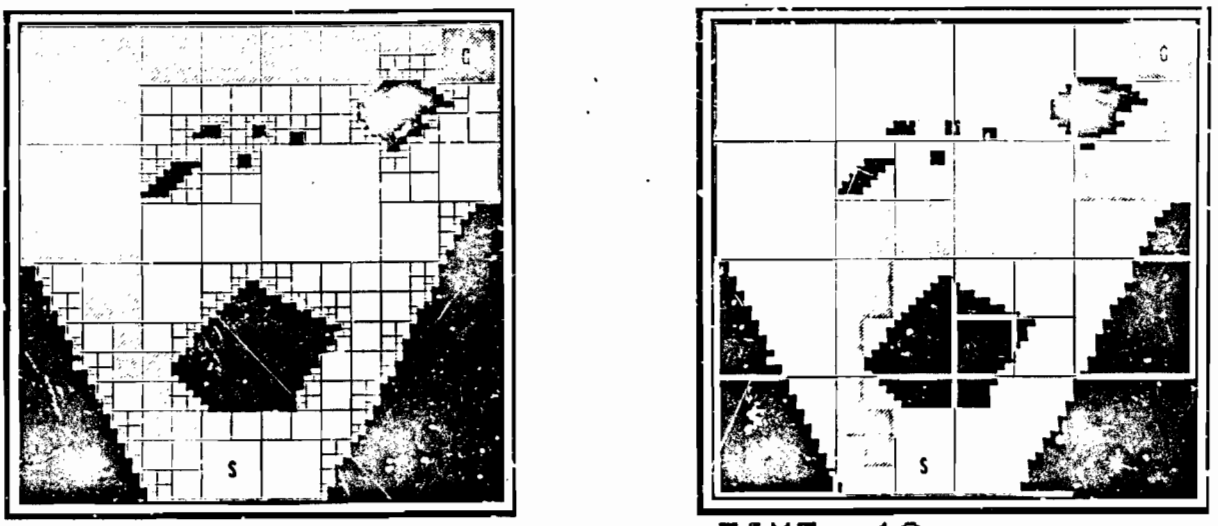
An example of single-stage planning on a pure quadtree



(b)

Staged path planning with leaf node thresholding as the pruned quadtree generation strategy

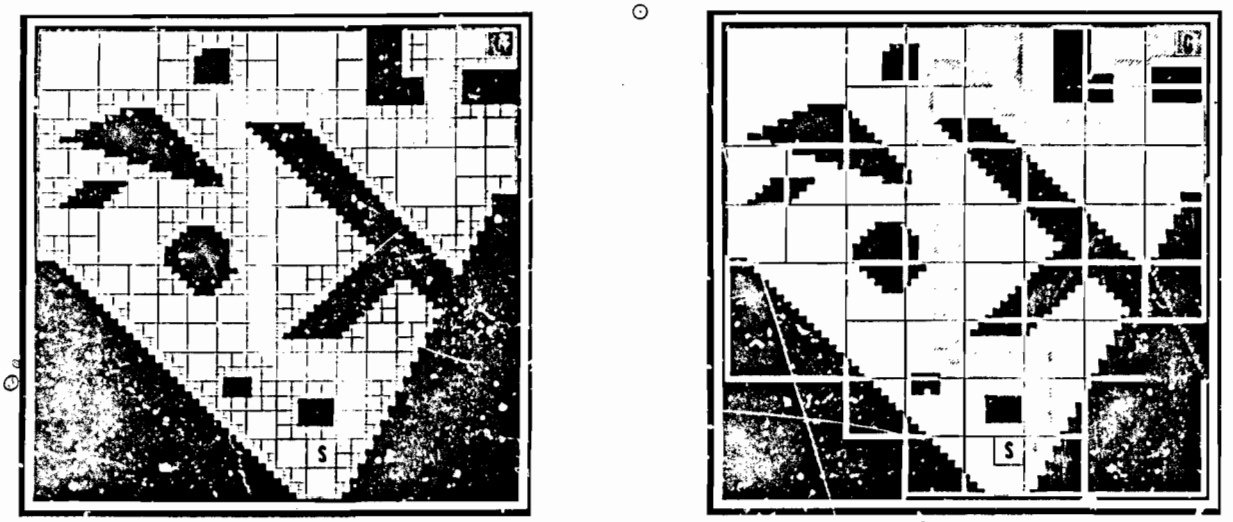
Figure 3.3 Examples of pruned quadtree generation strategies



(a) (b)

Figure 3.4(I) Staged vs. single stage path planning (example 1)

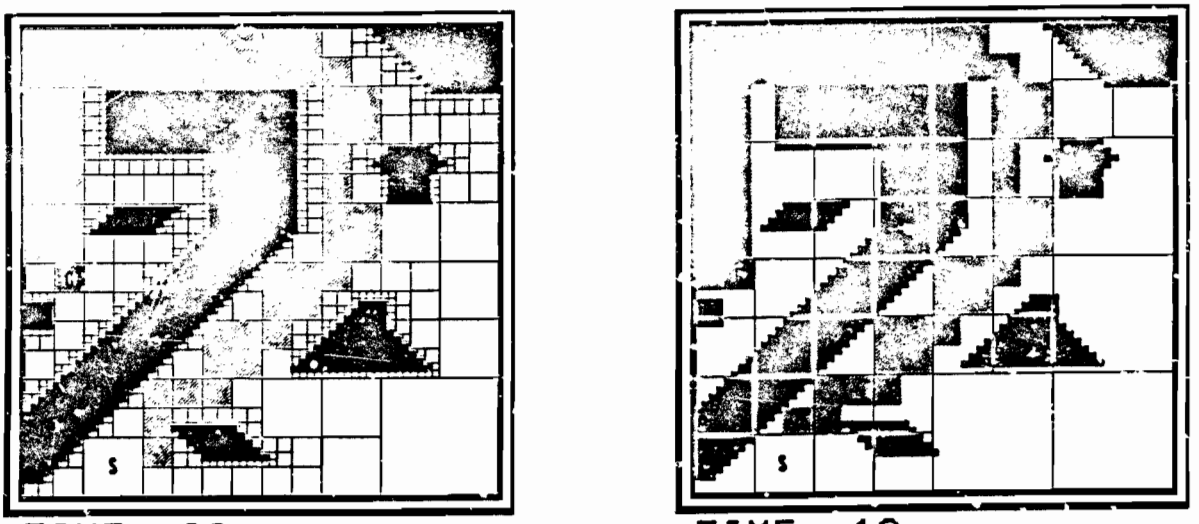
(a) shows the results of single stage planning and (b) shows the results of staged planning.



(a) (b)

Figure 3.4(II) Staged vs. single stage path planning (example 2)

(a) shows the results of single stage planning and (b) shows the results of staged planning.



TIME 38
EXPANDED 272/688
PURE A*

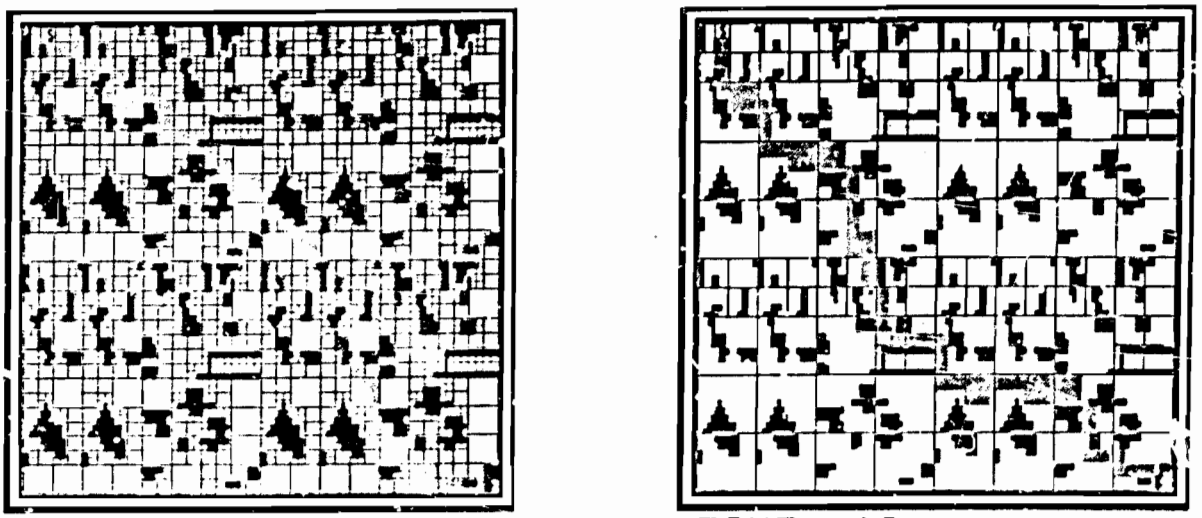
(a)

TIME 12
EXPANDED 29/40
LEAF-THRESH 50

(b)

Figure 3.4(III) Staged vs. single stage path planning (example 3)

(a) shows the results of single stage planning and (b) shows the results of staged planning.



TIME 168
EXPANDED 542/2863
PURE A*

(a)

TIME 18
EXPANDED 29/124
LEAF-THRESH 50

(b)

Figure 3.4(IV) Staged vs. single stage path planning (example 4)

(a) shows the results of single stage planning and (b) shows the results of staged planning

GC970145-46

ERROR DETECTION AND CORRECTION FOR STEREO

Rakesh Mohan
Intelligent Systems Group
Departments of Electrical Engineering
and Computer Science
Powell Hall Room 224
University of Southern California
Los Angeles California 90089-0273

ABSTRACT

Current line-by-line stereo algorithms make numerous wrong matches. We propose a method, based on figural continuity along linear segments, which detects and corrects a certain type of those errors for edge based stereo. This technique is extended to provide a quantitative measure of performance of such stereo algorithms. A few evaluation functions used for line-by-line stereo are compared using this method.

1. INTRODUCTION

The relative displacement in the position of objects, as viewed by a pair of eyes, is an important source of depth information for humans. This phenomenon, called binocular stereopsis, can be used by computers to locate objects in 3D. The difficulty faced in the implementation of stereo is the task of identifying corresponding locations in the two images. For a survey on stereo consult [1].

Our goal is to improve the performance of stereo matching algorithms. We take the case of line-by-line edge based stereo and propose two ways of improving their performance.

1. Given the output of a stereo algorithm, i.e. the sparse map of the disparities obtained at the matched edge points, we wish to detect and correct errors made by the matching program. We classify the type of matching errors into two categories. Using a strict geometric constraint on the distribution of disparities along linear segments, we are able to identify and correct one type of error. Also in the process, we are able to fill in disparities for edges which have not been matched and some occluded edges.
2. We wish to improve the performance of the stereo program before the correction phase. The quality of a matching algorithm can be judged by the amount of errors it makes. Being able to detect one type of matching errors, we compare the performance of various match evaluation functions which could be used by the stereo algorithm, and choose the one which gives the least amount of matching errors.

This research was supported by Defense Advanced Research Projects Agency (order number 5-N15-42 and 5N15-84 and monitored by Wright Patterson Aeronautical Labs.

In section 2, we discuss feature based stereo. In section 3, matching errors in line-by-line stereo are classified into two categories. A continuity constraint for disparity values is derived from geometric principles in section 4. The algorithm to detect and correct one of these types of errors is presented in section 5. A measure of performance for line-by-line stereo is defined in section 6. In section 7 we compare the performance of a few match evaluation functions. Conclusions and further research areas are stated in section 8.

2. FEATURE BASED STEREO ALGORITHMS

Most feature based stereo algorithms use edges as the feature primitive to be matched. Edges have an extent of one pixel. Using an epipolar geometric constraint for stereo, edges lying on an epipolar line are matched only to edges lying on the corresponding epipolar line in the other image. This type of stereo we term as **line-by-line stereo**. The main benefits of this technique are:

- The direct application of epipolar geometry forces a strong constraint on the search space.
- Global optimality has to be maintained only among edges on one epipolar line. These are much fewer than the edges in the whole image and hence, need less computation.
- All the epipolar lines can be processed in parallel.
- The ordering among edges on an epipolar line is obvious and natural. This is useful for dynamic programming based algorithms.
- All features (edges) are of equal length and the match along the epipolar line is across their full length. Matching of features thus does not have to be normalized for lengths or amount of overlaps.

Intensity discontinuities in images correspond to physical features in the imaged scene. These features could be surface boundaries or surface markings. These boundaries are connected, forcing the following **continuity constraint** on disparities:

> Disparity along a boundary changes smoothly, i.e. there should be no disparity discontinuities along a boundary.

In line by line stereo, when epipolar lines are processed independently of each other, the connectivity information contained in segments is not used and the continuity constraint is therefore not applied.

The following methods have been proposed to use the continuity constraint for line-by-line stereo

1 Use disparity information from already matched neighboring epipolar lines to guide the matching [2]. This technique is used mostly as a tool to cut computational costs by limiting the search space. The major drawbacks with this technique are:

   a Wrong matches in a line can be forced by wrong matches in the neighboring lines. As this line will now be used to guide matching in other lines, errors become cumulative.
   b The assumption made is that disparity changes **slowly** which is a stronger statement than saying that it changes smoothly. Also this assumption does not hold in general.
   c. Directional biasing. The results depend on whether the images are processed top to bottom or the other way around.

2 Use the continuity criteria after the complete matching process.
   For example

   a. Arnold [3] chooses a suboptimal match which meets the continuity criterion (based on some statistics) better than the optimal one.
   b. Adjacent edges having disparity difference greater than some value based on on some statistics signal an error and a cooperative process is used to detect and reject the wrong match by Baker [4].

   In these algorithms, the measure used does not follow from the geometric implications of continuity but relies on having nearly same disparity along a boundary.

3 Use continuity constraint for search in a 3D search space
   Ohta and Kanade [5] use dynamic programming in three dimensions for intra-scanline search to find consistency among scanlines. Their technique is computationally very expensive so only a few connected contours are used. It is also not clear if optimizing a path in the search space based on some cost will always maintain three dimensional continuity.

The above techniques do not propose disparities for edges not matched or rejected as incorrectly matched and for occluded portions of segments. Surface interpolation techniques might be used to fill in disparities but such methods do not use the continuity constraint along boundaries. The continuity constraint has also been used for other type of stereo algorithms for example [6] and [7].

Edges can be linked up into contours called **segments**. These contours can be of arbitrary shape to reflect the shape of the underlying feature or they can be linear approximations to it. The latter type of segments are termed as **linear segments** and have been used as feature primitive for stereo matching by Medioni and Nevatia [8].

The matching of segments incorporates boundary connectivity directly. Also, segments are more complex features than edges and thus contain more information which could be used to give more precise match evaluations than for edges. The segments can be as short as a single edge so segment based stereo does not break down in the presence of isolated edges [5]. On the other hand, such isolated edges should be treated with suspicion, given the connected nature of physical features. The problems inherent to segment based stereo matching algorithms are:

- The problem is in identifying those properties of segments which can be used in matching, and in deciding how these properties can contribute to the confidence we can have in the matches. Medioni and Nevatia [8] use segment orientation and contrast, and difference of disparities among neighboring segments for the evaluating a match.
- The use of epipolar geometry is not as straightforward as in line-by-line stereo. Although a match for a segment can be restricted to a window bounded by epipolar lines passing through its end points, we cannot ensure that a matching segment can be found which exactly fits this window. Matches for only parts of the segment are found. The matched segment often extends beyond the epipolar window. The lengths of a segment and its match may differ a lot. A segment may find more than one segment which it can match. These matching segments may be consistent in that they do not overlap along epipolar lines (i.e. along any given epipolar line, the segment in one image has only one match in the other image) but they may propose different disparities for the segment they match. All these issues complicate the task of evaluating a match. Many of these problems stem from the poor quality of segment detectors.
- Global optimality has to be found among all the segments in the image. The number of segments in an image is much larger than the number of edges along an epipolar line so insuring global optimality is more expensive.

3. ERRORS IN LINE-BY-LINE STEREO

It is our observation that in any line-by-line stereo, a lot of edges are assigned wrong matches. Wrong matches could be spurious null matches, or matches to a wrong edge. Edges might not be matched because the match evaluation function used by the stereo matcher assumes that the edge is spurious, or that its matching edge in the other image is missing or that the edge is occluded in the other image. Wrong edge matches are of the following two types.

Type I (local) errors:
In figure 1 the edge pairs matched by a stereo algorithm are shown linked by the epipolar line. The contours represent the segments detected in the image. The figure shows that more edges of the segment AB are assigned matches to the correct segment CD than to any single wrong segment. These wrong matches arise due to the

fact that information along a single row of pixels may not be good basis for matching and so there will be some wrong matches on an epipolar line. Also the real epipolar line could be locally distorted due to imaging device and conditions [6] or due to noise.

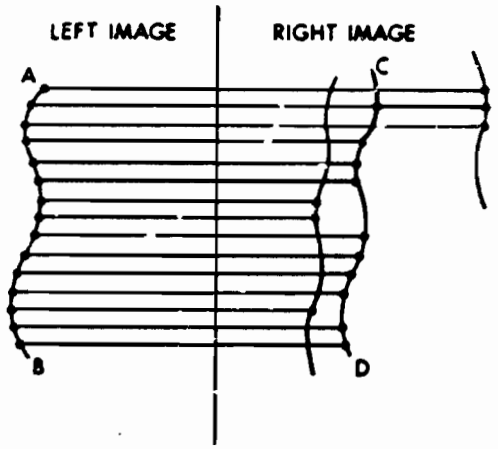


Figure 1: Type I error

Type I errors can be detected and corrected on the basis of the continuity constraint.

Type II (global) errors:
Figure 2 shows another type of wrong match. Here more edges of the segment AB are assigned matches to a wrong segment EF than to the correct segment CD. We term such errors as Type II errors. These errors reflect that the function for evaluating the quality of match used by the stereo algorithm prefers the wrong segment as a better match. This is a drawback of the match evaluation function. We do not believe that any single evaluation function can always avoid Type II errors.

Type II errors can not be detected (or corrected) on the basis of the continuity constraint. In fact, all the stereo algorithms which use figural continuity along segments, including segment based stereo, can directly deal with only type I errors.

This classifications of errors holds for all type of segments. However, from this point on, we will be dealing exclusively with linear segments.

The more information we use to evaluate the goodness of a match between two edges, the less matching errors we should have. However, using more information means increased computational expense and the benefits in terms of reduced error may not be proportional to the extra cost. Thus a cheap way to detect and correct errors would be useful in improving the performance of simple evaluation functions.

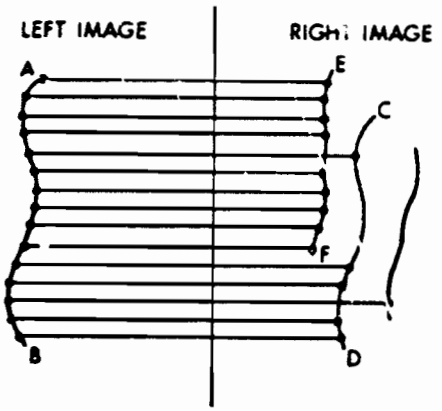


Figure 2: Type II error

Also, we do not know all types of information which could be used to evaluate edge matches. Even with the known metrics for judging the quality of a match, it not clear as to how much importance should be attached to different measures. Nor do we know much about the variation in performance of the measures due to changes in the images. This situation is reflected in the fact that the various line-by-line stereo algorithms in existence use different cost functions. One reason for this state of affairs is the difficulty in measuring quantitatively the performance of a stereo algorithm and so not much work has been done on the comparison of the performance of various evaluation functions. In section 7 we shall compare a few cost functions used for line-by-line stereo algorithms.

4. DISPARITY CHANGE ACROSS LINEAR SEGMENTS

Let us now consider the special case of linear segments. A linear segment in an orthographic or perspective projection of a 3D scene is the image of a linear feature (or a linear approximation of a feature) in the 3D scene ruling out accidental alignments. In a stereo image pair even such accidental alignments in one image will be revealed in the other image.

Depth changes linearly along a straight line in 3D. Consider figure 3 of the orthographic projection of a straight line in the scene onto the image plane (x-y plane) since

$$\frac{\quad}{\quad} = \frac{\quad}{\quad} \qquad (1)$$

we have

$$\Delta ABC \sim \Delta CDE$$

and

(2)

We also note here that it is obvious from the equation of a line

$$\frac{x-x_1}{x_2-x_1} = \frac{y-y_1}{y_2-y_1} = \frac{z-z_1}{z_2-z_1} \quad (3)$$

that the depth is linearly proportional to the displacement along the x-axis (or the y-axis).

In stereo, matching gives us the disparity, not the depth. In the image, the projection is perspective rather than orthographic. Therefore, we can not directly use the relationship derived in equation (2). Consider figure 4 showing a linear segment in a stereo pair. As is obvious from figure 4, disparity changes linearly along the linear segments irrespective of the type of projection, camera geometry etc. We will use this strong constraint to detect and correct wrong disparity values due to bad matches.

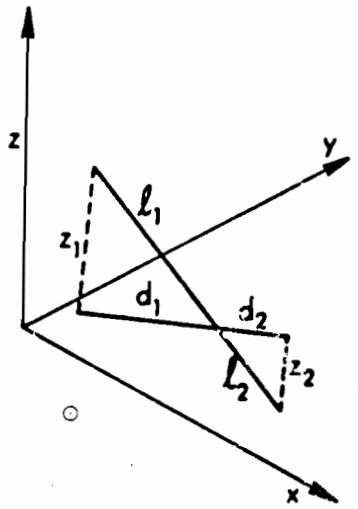


Figure 3: Orthographic projection of a straight line

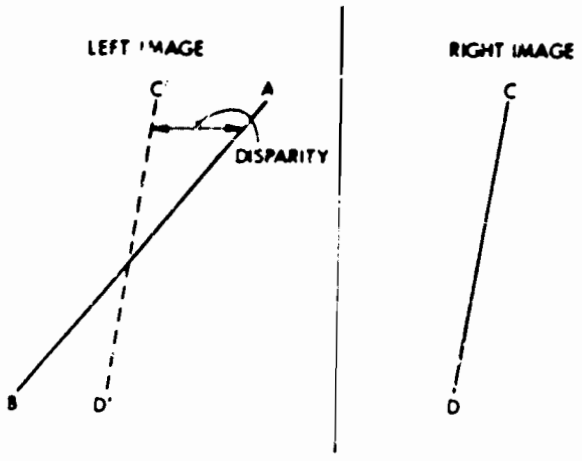


Figure 4 Matching linear segment in stereo pair

5. DETECTING AND CORRECTING TYPE I ERRORS

We shall use the connectivity constraint for linear segments to maintain inter-epipolar line consistency. In section 4 we showed that disparity along a linear segment varies linearly. We use this strong constraint to detect Type I errors.

We work in a length-disparity (l-d axis) coordinate frame. First a linear segment is picked. The disparities obtained by the stereo matching program for the edges belonging to this segment are plotted as a function of the distance of the edge from one end of the segment. Since disparity varies linearly along a linear segment, if all edges are matched correctly, all the points plotted should fall on one straight line, as shown in figure 5.

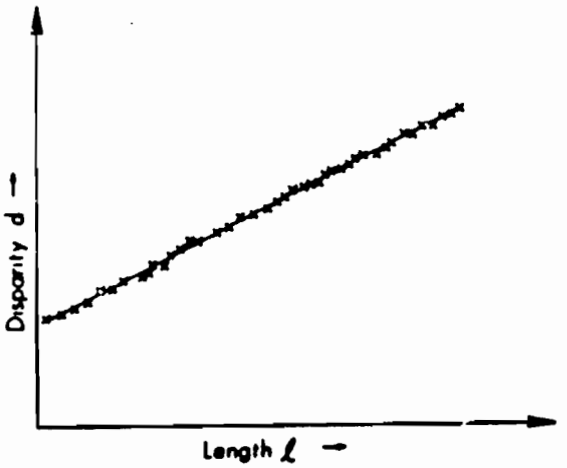


Figure 5: Plot of ideal disparity versus length on a segment

Figure 6, however is a more typical output. If we can interpolate a line through this data which would correspond to the correct disparity plot, we could not only detect and correct wrong matches but also fill in disparity information at edges which were not matched and to edges belonging to occluded portions of the segment.

A traditional method to fit a straight line through experimental data with errors is to interpolate a line which minimizes the square of the errors. However, we cannot directly apply this technique here because the least squared error line is valid only when the distribution of errors about the correct values is gaussian. In the plot of disparities along a segment this does not hold. Since a wrong match means that the edge was matched to an edge belonging to a segment in the neighborhood of the correct segment the distribution of wrong matches are usually strongly biased to one side of the correct value. For example if the segment has high disparity then most of the wrong matches give a low disparity (and vice versa). Also because of the spacing between the segments the amount of error in disparity caused by a wrong match is usually large. As the least squared error line uses the square of the errors for interpolation even a few

wrong matches used for interpolation could pull the line substantially off the correct position. Fischler and Bolles [9] have presented a technique to deal with erroneous data. We will use another method more suited to our task domain.

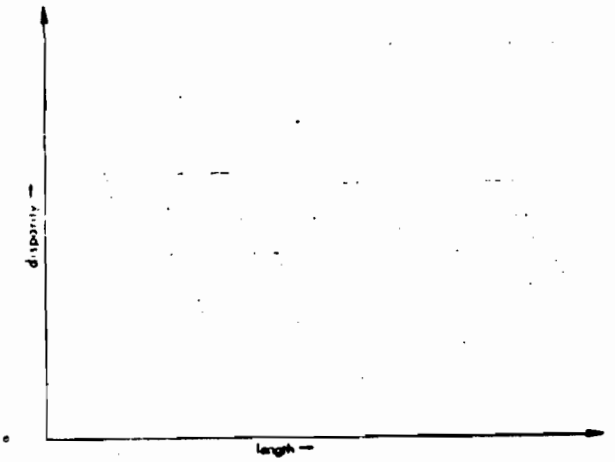


Figure 6: Plot of real disparity versus length on a segment

We have to throw away the points plotted due to wrong matches before we can interpolate a line through the disparity values. If more edges are matched to the correct segment than to any other single wrong segment (Type I error) then we can detect the errors in the following way. We fit thin strips of all orientations and locations to the points. From the definition of type I errors it follows that the strip which has the maximum number of points in it has has the disparity values corresponding to the correct matches, the rest are erroneous. The reason for choosing thin strips instead of straight lines is that due to the discrete location of the edges and the error in the location of edges, all the points from correct matches do not fall exactly on a line.

To fit the thin strips to the data we use a version of the modified Hough transform technique proposed by Wallace [10]. Consider figure 7. The window ABCD of the d-l place contains all the plotted disparities. The thin strip fitted on the correct data points has to intercept AB and CD. We choose the intercepts t_1 and t_2 made by the line running through the center of the strip at AB and CD as the parameters for the Hough transform. The width of the strip will depend on the size and type of filters used for edge detection. If the edge detector can not resolve between two edges less than **p** pixels apart then the width of the strip has to be less than 2**p**. We use a strip three pixels wide and we allow an overlap of a pixel between two adjacent parallel strips. We will then have to consider the intercepts only at every other pixel along AB and CD. Say the range of disparity in the image is **d**. Then the size of the grid for the Hough transform is **d** $\times$ 4.

Once the correct matches have been identified we can get rid of the gross disparity errors through their disparities from this line we can get the disparities of all the edges on the segment. These new disparities are stored as real numbers since we can get subpixel disparity values. We also store the disparities at the begin and the end points of segments for display purposes. This process is repeated for all linear segments in the image.

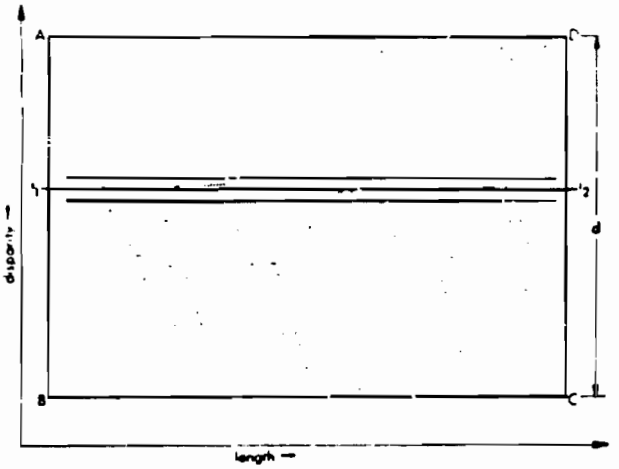


Figure 7: Thin strip containing the correct disparities

At this stage, we are not able to detect type II errors. Information from other sources which provide depth information, such as monocular cues, shape from shading, smoothness constraints on surfaces, another stereo matcher etc. might indicate that the disparity obtained at some of the segments is wrong, thus detecting type II errors.

6 QUANTITATIVE COMPARISON OF MATCHING ALGORITHMS

It will be very useful to have a quantitative measure of the performance of a stereo algorithm. This measure could be used during the development phase of a stereo algorithm. Say for example a line-by-line stereo algorithm is being developed and the match evaluation function uses some weights and thresholds. The measure of performance could be used to fine tune the weights and thresholds for a particular class of images. Better yet if the performance measure could be calculated cheaply and automatically, it could be used to dynamically change the weights and thresholds as different parts of an image are processed.

Some stereo algorithms are designed for specific image classes like rolling terrains, while others are designed to be general. Even a general purpose stereo algorithm may exhibit a bias towards some particular type of imagery due to assumptions made in its design (smoothly varying surfaces, no order reversals etc.). Applications of stereo usually use a certain type of images, for example, mapping would use aerial imagery, while robotic work would use indoor scenes photographed at short distances. Using a performance measure, an installation could choose from available stereo algorithms the one best suited for its task domain.

An obvious performance measure would be the percentage of matching errors and the percentage of edges matched. The percentage of edges matched could be easily obtained from the stereo algorithm itself. Used alone it could be misleading as it would not be known how many of those matches were incorrect. To calculate the percentage of errors, we need an algorithm to consistently identify the wrong matches. As we have seen in section 5, we can locate Type I errors consistently.

The algorithm to calculate the percentage error is:

(1) Link the edges into linear segments.
(2) Plot the disparities obtained along the length of the segment.
(3) Fit thin strips to them to identify the correct matches.
(4) Accumulate the number of correct and incorrect matches.
(5) Repeat steps 2 to 4 for each segment.
(6) Percentage error = 100*wrong-matches/total-matches.

One might question the need of evaluating the error percentage since this algorithm not only detects errors but can also correct them. We still need to evaluate the performance since

1. The more points we have on a linear segment with the correct disparity, the more accurately we can interpolate the disparity for the full segment. Therefore, even with the error correction, we would prefer low error stereo algorithms.
2. This technique does not work for very short segments due the small number of points available. We cannot correct the matches on such segments. The smaller the overall matching error is, the more confidence we can have for matches for edges along these segments.
3. We can detect only Type I errors by this method. Type II errors are also a serious problem in stereo. However if more edges on a segment with Type I error were matched incorrectly it might have deteriorated into a Type II error. So a large percentage of Type I errors is also a good indication that a lot of Type II errors are also present.
4. The performance of area based stereo algorithms is usually the poorest at intensity discontinuities. This method can give an evaluation of their performance at the edges (which might not hold for their performance away from edges).

There are some obvious limitations of this evaluation algorithm. It cannot measure the amount of Type II errors. Because of this, it cannot be used to evaluate the performance of stereo algorithms which match segments [8] (or more complex features).

7. COMPARISON OF A FEW COST FUNCTIONS FOR LINE-BY-LINE STEREO

We take a typical line-by-line stereo algorithm using dynamic programming to compute the best matched edge sequence along an epipolar line i.e. the sequence which has the minimum total cost of matching edge pairs. The algorithm chosen is one implemented by Ohta and Kanade for their intra-scanline search and details can be found in [5]. We evaluate the matching performance using the match cost function proposed by them against a modification of this function and two other functions.

Table 1 shows the performance of this algorithm using four different match evaluation functions on the stereo pair in figure 8. The edges and the linear segments have been detected using Nevatia and Babu's linear segment extraction algorithm [11]. Figure 9 displays the disparities along those segments in the left image that have been processed by the error detection and correction phase.

The following are among the measures used for the evaluation:

- Total number of segments selected:
  We use segments of length greater than a set limit to do error detection.
- Total number of segments processed:
  From the selected segments, only segments having more than a fixed number of matched edges, and correct matches, are used for the error statistics (and for correction) to ensure reliability.
- Percentage matched edge points processed:
  This is a percentage ratio of the matched points which were on the processed segments to all the edges matched.
- Percentage error
  This is the percentage of matched points which were in error among the matched points processed.
- Percentage edge points corrected.
  Number of edge points which were either corrected or had their disparities filled in from all the edge points on the processed segments.

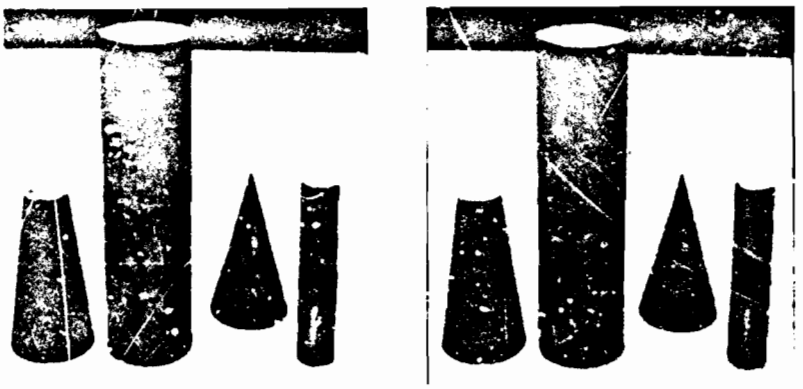
The match evaluation functions compared were

- Cost Function I:
  This is the cost function used by Ohta and Kanade and its details can be found in [5].
- Cost Function II
  To function I we add the restriction that matches are considered only among edges whose direction does not differ from each other by more than 30.0 degrees. The directions used used are the orientation of the segment the edge belongs to, and not local edge orientations.
- Cost Function III
  This function is formulated to favor matches among edges with similar orientations and with similar interval lengths on their left between them and their preceding match.
  $cost = \exp(|\theta_1 - \theta_2| / c_1) \; c_2 * (|\alpha_1 - \alpha_2| / (\alpha_1 + \alpha_2)^2 * c_3)) \; (\alpha_1 + \alpha_2)$
  Where θ are the edge orientations, α the interval

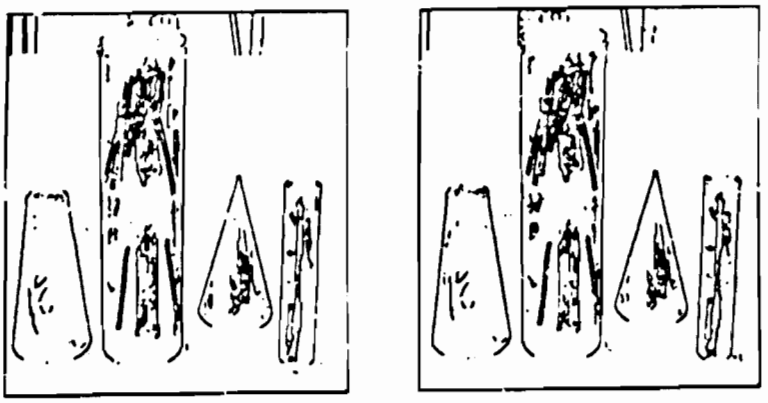
lengths and c_i constants

- Cost Function IV

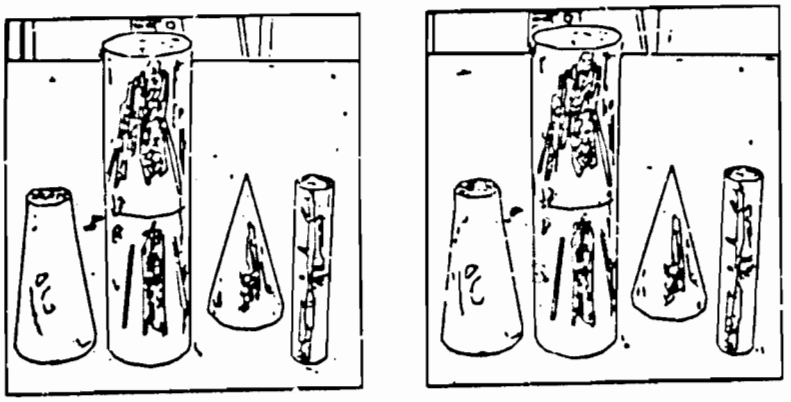
  For this function we use the intervals lying on the left of the two edge points. The cost of matching the two intervals is the best cost of matching them using an area based stereo algorithm. The area based stereo algorithm used is one proposed by Leguilloux [12].



(a) intensity



(b) edges



(c) linear segments

Figure 8: Image I

	COST FUNCTION I	COST FUNCTION II	COST FUNCTION III	COST FUNCTION IV
% edges matched	88.83	88.38	58.87	78.88
% segments selected	38.46	38.46	38.46	38.46
% segments processed	24.14	23.86	16.33	14.31
% matched edge points processed	83.86	78.28	66.86	53.87
% error	3.88	3.83	8.84	11.88
% edge points corrected	14.81	15.13	46.81	46.82
maximum % error for a segment	38.88	27.27	58.88	81.84

Table 1: Performance of cost functions on image I

	COST FUNCTION I	COST FUNCTION II	COST FUNCTION III	COST FUNCTION IV
% edges matched	77.52	74.81	56.86	70.53
% segments selected	37.31	37.31	37.31	37.31
% segments processed	28.15	28.15	14.18	11.84
% matched edge points processed	76.38	76.77	54.16	33.19
% error	6.81	7.52	17.81	38.45
% edge points corrected	14.85	16.11	41.11	42.86
maximum % error for a segment	46.86	44.44	56.82	82.17

Table 2: Performance of cost functions on image II

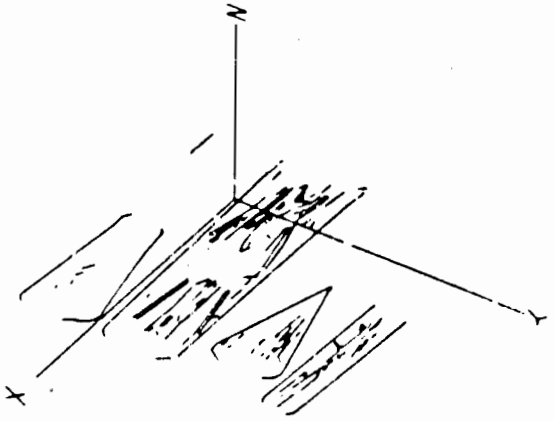


Figure 9: Disparity values at processed segments

The performance of these algorithms for the image pair in figure 10 are listed in table 2. Figure 11 shows the disparities along the processed segments of the left image.

In the comparison of the four match evaluation functions, it is clear that functions I and II perform appreciably better than functions III and IV, but among functions I and II it is difficult to choose the winner.

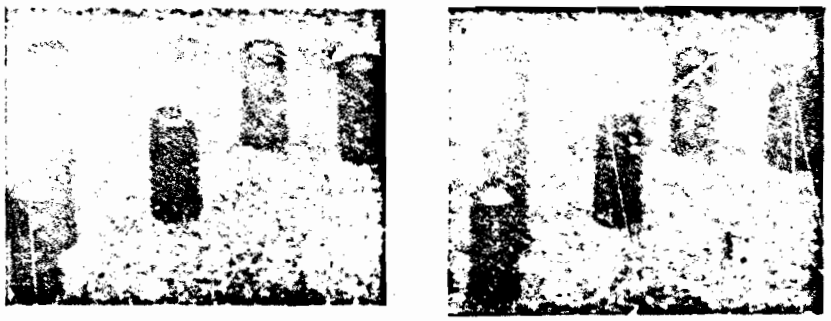
8. CONCLUSION

The proposed algorithm is linear in the number of edges processed. Therefore, it is a very cheap way of insuring figural continuity along segments. Although, the algorithm does not use the continuity constraint during the matching process, we believe that if a substantial number of the edges can be processed by this algorithm, we can afford to ignore the other matched edges. Even if we use the corrected disparities as fixed constraints for a second pass over the image pair, the algorithm would both be computationally cheaper and would use more continuity constraints than [5].

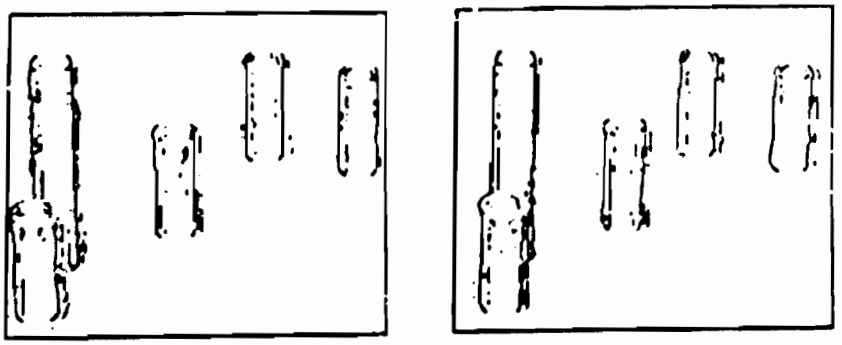
We have been able to demonstrate that we can process a large number of the matched edges and detect type I errors in them. We can also correct these errors and fill in disparities for edges not matched. However this algorithm is weak in the following areas:

- Not all matched edges are processed. This is due to the fact that at each segment we need some minimum number of correct matches before we can confidently interpolate disparity for the whole segment.
- The problem of type II errors has not been addressed.

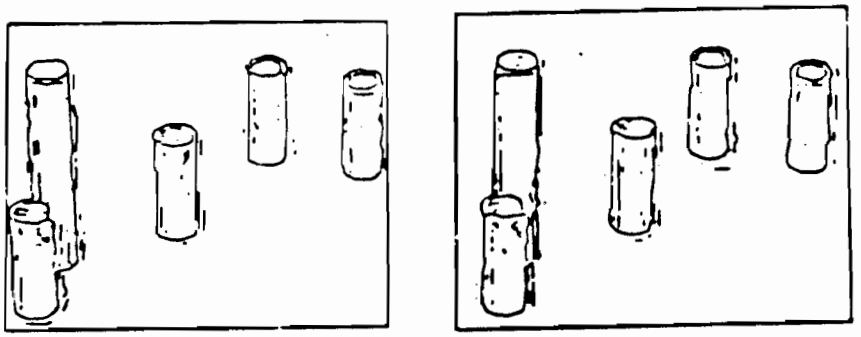
The next step in this research will be to use the corrected disparities as input constraints for reprocessing the stereo pair using the same or a different stereo algorithm.



(a) intensity



(b) edges



(c) linear segments

Figure 10. Image II

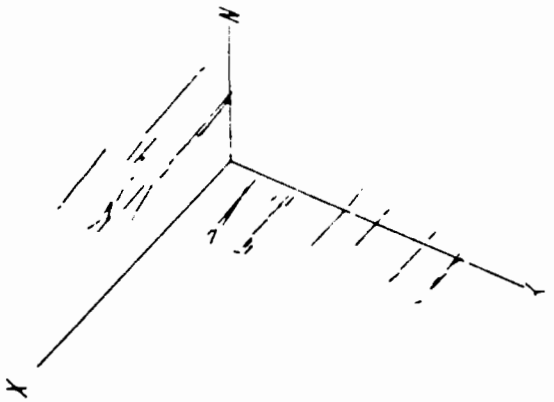


Figure 11: Disparity values at processed segments

References

1 Barnard, S and Fischlar M. "Computational Stereo," *ACM Computing Surveys*, Vol 14, No 4, December 1982, pp 553-572

2 Henderson, Robert L, Miller, Walter J, and Grosch, C B "Automatic Stereo Recognition of Man-Made Targets," *Society of Photo-Optical Intrumentation Engineers, Vol 186, Digital Processing of Areial Images*, August 1979.

3 Arnold, R. "Automated Stereo Perception," Tech report No STAN-CS-83-961, Stanford University, Computer Science Department, March 1983

4 Baker, H. "Depth From Edge and Intensity Based Stereo," Tech report No STAN-CS-82-930, Stanford University, Computer Science Department, September 1982

5 Ohta Y and Kanade T "Stereo by Intra and Inter-scanline Search Using Dynamic Programming," Tech report CMU-CS-83-162, October 1983

6 Grimson, W E L, Computational Experiments with a Feature Based Stereo Algorithm," *Pattern Analysis and Machine Intelligence*, Vol 7 No 1, 1985.

7 Mayhew, J E W and Frisby J P, "Psychophysical and Computational Studies towards a Theory of Human Stereopsis," *Artificial Intelligence*, 1981, pp 349-385

8 Medioni, G and Nevatia R, Segment-based Stereo Matching, *Proceedings of DARPA Image Understanding Workshop*, Washington, D C, June 1983

9 Fischler, M A and Bolles, R C Random Samples Consensus A Paradigm for Model Filtering with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, Vol 24 No 6 June 1981 pp 381-395

10 Wallace, R S, "A Modified Hough Transform for Lines," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco June 1985

11 Nevatia, R and Babu, K R, "Linear Feature Extraction and Description," *Computer Graphics and Image Processing* , Vol 13, 1980, pp 257-269

12 Le Guilloux, Yann, *Determination automatique du mouvement dans une sequence d'images. Interet pour l'interpretation*, PhD dissertation, Ecole Nationale Superieure Des Telecommunications, June 1984

GC970145-47

GEOMETRIC GROUPING OF STRAIGHT LINES

R. Weiss, A. Hanson, E. Riseman
Computer and Information Science Department
University of Massachusetts
Amherst, Massachusetts 01003

Abstract

This paper presents a new approach to the extraction of straight lines based on geometric grouping. Zero crossing points of the Laplacian of intensity images and the gradients at those points are used. These edges are input to a hierarchical linking and merging algorithm. Edges are linked based on both intrinsic and geometric properities, e.g. if their gradients are similar, their contrasts are similar, and their endpoints are close. In the merging process, if a sequence of linked edges can be approximated sufficiently well by a straight line, then they are grouped and are replaced by longer straight lines. The hierarchy allows us to represent lines at multiple scales, but does not involve smoothing the image. There are four advantages of this approach for extracting straight lines: 1. It links collinear line segments even when they are separated by gaps. 2. Low contrast lines may be found as easily as high contrast lines. 3. It is less sensitive to texture than zero crossing contours when extracting boundary lines. 4. It can find lines which may not be straight locally but are straight at a larger scale.

1.0 INTRODUCTION

The extraction of lines based on either significant intensity changes or perceived boundaries between areas is a difficult and important step in image understanding. This paper presents a new approach to the extraction of straight lines based on geometric grouping. The primary goal is the extraction of straight lines from images in which there are fragmented intensity discontinuities. The secondary goal is the demonstration that the use of geometric organisation can be an important part of the line extraction process and therefore can produce improvements when combined with standard edge detection techniques.

Our view of the task of image understanding is that it is based on a process of organising events in an image or sequence of images into structures which can be matched with models of objects in the physical world. A postulate of this paper is that this organisation process uses both geometric and intrinsic properties of structures in the image. In particular we apply this to straight lines.

A straight line is not just a local event. Figure 1 shows some examples of image events which are straight lines at some scales but not at others. In this paper, a straight line is defined geometrically by the property that it is composed of a sequence of line segments which are approximately colinear and that each segment is close to its successor. Both of these criteria depend on scale; long lines, for example, can be separated by a larger gap than small ones and still be close. The intrinsic property which defines straight lines is that the intensity gradients must be similar in magnitude and direction along the line. There

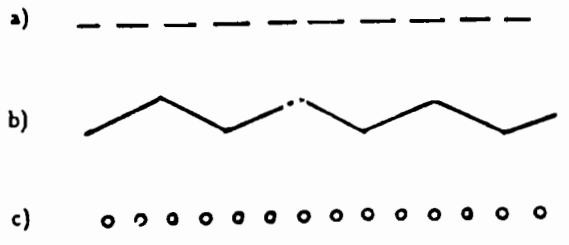


Figure 1. Events which can be perceived as a straight line.

are other possible definitions of straight lines, which would lead to different results. For example, Figure 1c shows dots which are perceived as a straight line, but not on the basis of gradient information.

In Section 2, we describe the algorithm used to find straight lines as we have defined them and in Section 3 we present some of the experimental results.

2.0 DESCRIPTION OF THE ALGORITHM

The two major components of the algorithm for extracting straight lines are edge detection and hierarchical grouping. Hierarchical grouping has two steps which are performed at each level: linking and merging.

2.1 Edge Detection

There are many edge detection algorithms which might be used. The main requirements are that the algorithm produce measurements of the intensity contrast and direction of the edge. The two algorithms which we have used for selecting points are zero crossings of the Laplacian operator and a directional edge operator based on the work of Haralick and Canny [2,4]. Although there may be algorithms which have better performance in some cases, the ones chosen are representative of a class of algorithms; most of them are expected to have the same types of problems which we encounter here [3].

Calculation of initial edges

Since most of the results have been obtained for the Laplacian operator, we describe the processing for that one in more detail. First, the image is convolved with a 3x3 mask which approximates the Laplacian:

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

The advantages of using the Laplacian operator are:

1. High positional accuracy of an edge, even with aliasing.
2. Good sensitivity to high frequency data.
3. Reduction of the data. Many pixels don't produce edge points.

Next, zero crossings of the Laplacian image are detected and their positions determined by linear interpolation. An edge, which is a line of length 1, is positioned with its center at a zero crossing. If a pixel value is zero, the adjacent pixels are checked, so only true zero crossings are considered. The orientation of the edge is determined by the gradient at the midpoint of the edge, i.e. the edge orientation is perpendicular to the gradient. Each edge carries with it a magnitude and direction (signed gradient magnitude) and each edge (now considered as a line) has an initial start point and a final end point.

When the Laplacian surface looks like a saddle point near the zero crossing set, then this set will look like a pair of hyperbolas or a pair of lines which cross. Experiments were performed in which 4 edges were created at these points to act as "continuation" edges for those around the saddle point. It was found that saddle points could be ignored, since the geometric context provided enough information to bridge the gaps when those points were omitted.

The most important difference of our approach from the usual one, which is to follow the zero crossing contour, is that it uses the gradient information at the edge point. Since the Laplacian operator only depends on second order terms, the tangent to the zero crossing contour is independent of the gradient.

Ordinarily there are several sources of problems with the Laplacian. Subsampling of an image and the presence of noise can result in isolated spots which have a high contrast with their neighbors. As a result, the zero crossing contours surround these isolated pixels and produce a point-like structure. Noise can introduce numerous local maxima in the gradient of the intensity, producing multiple edges parallel to the visually significant one. In addition there are zero crossing of the Laplacian where the gradient has a local minimum. These are called anti-edges, and the current implementation doesn't test for this event. Many of these problems can be solved by smoothing with different width

gaussian masks. Witkin and others have explored this using scale space [5,6]. The problem with smoothing is that it removes details which are important for some structures. For example, a very thin line, even if it is long, will become undetectable if high frequency data are filtered out. Our approach is to use the geometric context to eliminate edges which do not form lines in the image.

Directional edge operators

As an alternative, the zero crossings of an operator described by Haralick have been used instead of the Laplacian to determine the positions, where gradients should be sampled. The Haralick operator applied to an image with intensity function I is

$$\nabla(\nabla I)^2 \cdot \nabla I$$

We performed experiments in which the gradient operator was approximated by a 1x2, 2x2, and 3x3 masks. We did not use an approximation of the data by polynomial function. Since an image is given by discrete data, it is possible for the gradient to change direction completely without changing magnitude, so this operator would not detect such high frequency edges. An example is shown in Figure 2. There were such problems with high frequency data using the Haralick operator, consequently most of our experiments were done with the Laplacian.

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Figure 2: high frequency edges

2.2 The Hierarchical Grouping Process

The goal is the grouping of local edges or lines into longer lines, which implicitly defines a scale-space hierarchy. At each level, starting with edges which are one pixel, first linking is performed and then merging to produce lines in the next higher level in the hierarchy.

Linking

Linking of straight lines is a search for almost collinear pairs which are close to each other. The goal of this process is to reduce the search for candidate sets of lines for merging into longer lines. Lines within a *linking radius* of the endpoint of a given line are tested before linking. There are four criteria used for linking:

1 Similar gradient magnitude. The gradient magnitudes must be close to each other. The current system uses a factor of 2 as the test.

2 Approximately collinear. We required that the directions of the two lines must be within 30 degrees of each other, and that the distance between the midpoint of the second line and the first line be small. Lines 180 degrees apart are not linked.

3 Endpoints must be close. This is measured by the projection of the endpoint of one line segment onto the other line. The measurement need not be isotropic, and the criterion will depend on the scale.

4 Lines do not overlap. The final point of the first line must be closer to the initial point of the second than it is to the final point of the second.

One can view the result of the linking process as a directed graph with the line segments as the vertices and the links as the arcs. In general, a line will be linked with many other lines. The merging process examines paths in this graph and tests them for straightness. The linking process effectively reduces the number of combinations of lines which must be examined by the merging process.

Merging

The merging process consists of grouping and replacement, and incorporates the geometric context. The amount of context used is the *search radius*, which bounds the length of a sequence of lines which is grouped and tested for straightness. This sequence of lines is approximated by a straight line, and if the approximation is good, then a subsequence is replaced by a straight line at a larger scale. The algo-

rithm proceeds by examining each line in the link graph and performing the following steps:

1. If the line has already been merged with others, skip it.
2. Generate all paths of lines linked to the initial point or the final point, within the *search radius*.
3. Generate all paths which are combinations of paths from step 2: a path to the initial point, the line itself, and a path from the final point.
4. Fit a straight line to the set of endpoints, and measure the closeness of fit with the metric:

$$\frac{\sum d^2 \cdot w_i}{(n-2)\cdot s^2} \leq Const$$

where d is the distance from the endpoint to the approximating line, n is the number of endpoints, w_i are relative weights based on length, and s is the distance from the initial point of the first line to the final point of the last. In addition, there is a curvature measure, which is the reciprocal of the radius of the circle which is the best fit to the set of endpoints. This curvature is weighted by the sum of the lengths of the lines because the curvature is more significant for long lines than for short ones. The curvature is used to filter out curved lines.

5. Choose the path with the minimum error as measured above. Using the *replacement radius*, a subsequence of the path is replaced by a straight line whose gradient magnitude is the weighted average of those of its parts. Other paths through the original line are also replaced, but we require that there be at least a 40 degree difference in slope from one which has already been added. A single line may be copied to the next larger scale if it is long enough.

The hierarchical representation

The hierarchy consists of multiple planes, each at a different scale. A plane is a set of directed line segments together with a gradient magnitude. The scale of a plane is the range of the lengths of lines which can be stored in that plane. In the current implementation these ranges overlap. Thus lines which are not straight at one scale can still pass the straightness test at the next larger scale. Each plane is divided up into a grid corresponding to the *linking radius* in order to reduce the search computation during the linking step. Although all of the planes share the same coordinate system, there is no direct connection between the individual lines and the pixels of the original image. The grid is dependent on scale so that for any given plane the number of lines in a grid element should remain roughly the same over all scales. This is also based on the assumption that after linking and merging, the number of lines will be reduced by a constant factor. The hierarchy has 4 features:

1. It reduces the search space for sequences of lines to be linked and merged by making the *search radius* small at small scales.
2. It reflects the observation that "closeness" of lines is scale dependent.
3. It allows for a multi-scale representation of a line which may be straight only at large scales.
4. It is a compact representation.

3.0 EXPERIMENTAL RESULTS

The algorithm has successfully been applied to many images. Two of these are shown in Figure 3. As describe above, the first step is the computation of the zero crossings of the Laplacian. Figure 4 shows the zero crossings for the image in Figure 3a. As one can see, there are several places where the contour does not follow the boundary of the roof but branches off into the texture of the roof or the texture of the trees. Figure 5 shows the filtered output of the Burns straight line algorithm [1] with lines of length greater than or equal to 5. His algorithm is very successful in locating many of the straight lines, but some of them are fragmented. One can also find instances where the Burns

algorithm produces multiple parallel lines in a slow gradient, while the geometric grouping algorithm here does not. However, the problem of multiple parallel lines could occur and would require the use of two dimensional merging as a solution. Figure 6 shows the unfiltered output of the geometric grouping algorithm. By filtering this output to keep only the long lines, one can extract those straight lines which are likely to be significant, for example as boundaries of objects in the image. The results after filtering are shown in Figure 7. Figure 8 shows the results similarly obtained for the image in Figure 3b. Experiments have also been performed with aerial images, and the results are very encouraging. Figure 9 shows the results for a car with few long, straight lines. In the figure, the thickness of the lines is proportional to the contrast.

So far the only problem which we have encountered is overlinking of lines. As one can see in the roof in Figure 6, edges in the texture which have gaps but are collinear are sometimes linked and merged. In this case the gaps between the lines are important. Here the algorithm may be improved by using the density of lines to inhibit linking when the density is high. In our implementation, the *linking radius* depends on the scale, but it should really depend on the density of lines. We can see this in our own perception. If there is a high density of lines with different directions, we would only perceive a straight line if the gaps between the fragments were very small. Nevertheless, there will be cases where small gaps are important, and it will be necessary to consider other geometric context than just collinearity.

4.0 CONCLUSION

The results shown in this paper indicate that the use of geometric grouping in extracting long, straight lines produces a significant improvement over results obtained from standard edge detection algorithms. The linking and merging of straight lines into longer straight lines is a paradigm for the general process of linking and merging of geometric structures into larger and more abstract structures. It is clear from the way in which this works for straight lines that it is naturally a hierarchical process. An object may have different representations at different scales or levels of abstraction. In addition, the efficiency of the computation is dependent on hierarchical processing. Lastly, the geometric context to be used in computation is also a function of the level in the hierarchy.

References

1. Burns,J.B.,Hanson,A., Riseman,E., "Extracting Straight Lines", *7th int. Conf. on Pattern Recognition*, pp 482-485, Montreal, 1984.
2. Canny, J.F., "Finding Edges and Lines in Images", A.I. Lab MIT Tech. Report AI-TR-720, 1983.
3. Davis, L., "A Survey of Edge Detection Techniques", *Computer Graphics and Image Processing*, v.4,pp 248-270, 1975.
4. Haralick, R.M., "Digital Step Edges from Zero Crossings of Second Directional Derivatives", *IEEE Trans. Pattern Anal. Mach. Intell.*, v.6, pp 58-68, 1984.
5. Witkin, A. "Scale-Space Filtering",*Proceedings of IJCAI*, pp 1019-1021, Karlsruhe, 1983.
6. Yuille, A.L., and Poggio, T., "Scaling Theorems for Zero-crossings", MIT A.I. Memo 730, 1983.

Acknowledgements

This presentation is based on the work of Michael Boldt, who is currently a graduate student at the University of Massachusetts. We would also like to thank Brian Burns for his helpful ideas. Funding for this research was provided by DARPA grant N00014-82-K-0464.

(a) (b)

Figure 3. Two digitised images used to demonstrate algorithm

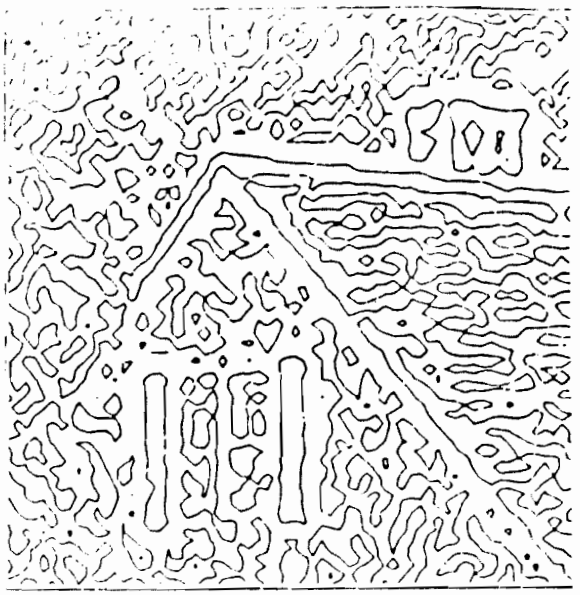


Figure 4. Zero crossings of the Laplacian for a part of Figure 3a

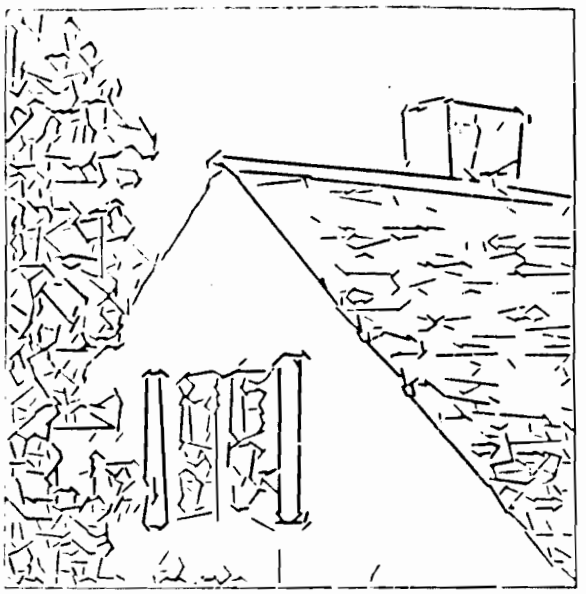


Figure 5. Filtered output from the Burns algorithm length ≥ 5

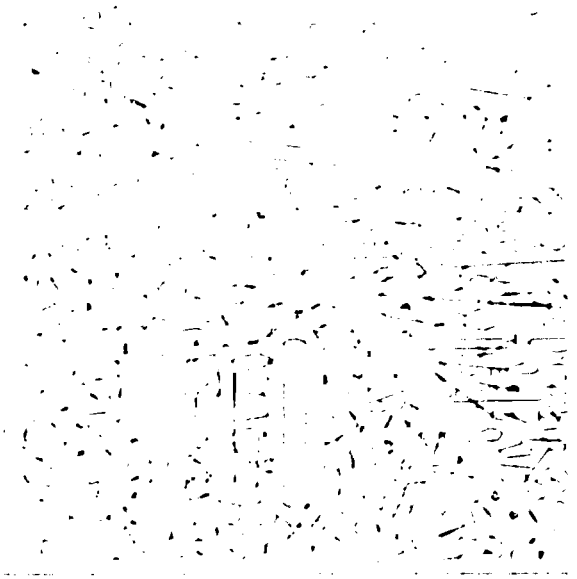


Figure 6. Output from geometric grouping algorithm for a part of Figure 3a

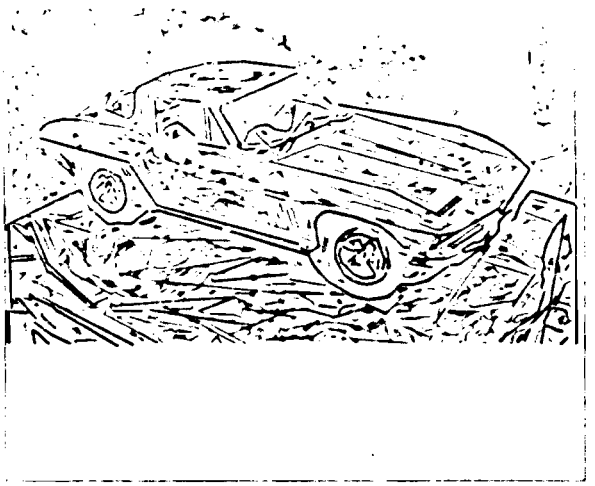


Figure 8. Filtered output from geometric grouping algorithm

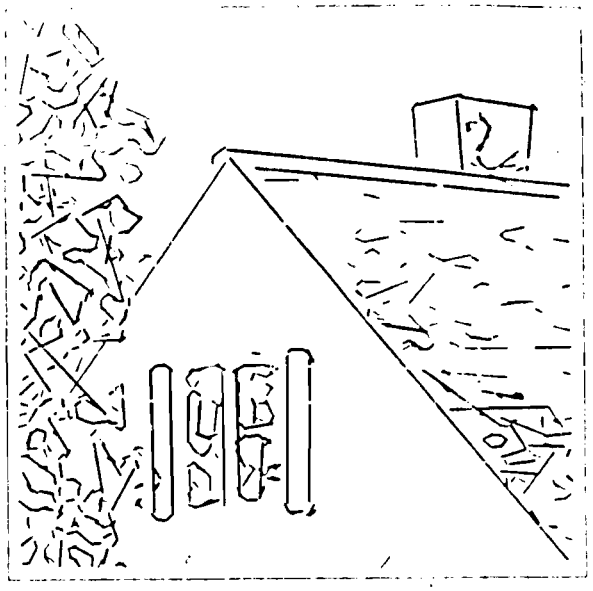


Figure 7. Filtered output from geometric grouping algorithm

GC970145-48

On Detecting Edges

Vishvjit S. Nalwa
Thomas O. Binford

A.I. Lab., Stanford University, CA 94305

Abstract

An edge in an image corresponds to a discontinuity in the intensity surface of the underlying scene. It can be approximated by a piecewise straight curve composed of edgels, i.e. short, linear edge-elements, each characterized by a direction and a position. The approach to edgel-detection here, is to fit a series of one-dimensional *surfaces to each window (kernel of the operator) and accept the surface description which is adequate in the least squares sense and has the fewest parameters. (A* one-dimensional *surface is one which is constant along some direction.) The tenk is an adequate basis for the step-edge and its combinations are adequate for the roof-edge and the line-edge.*

The proposed method is robust with respect to noise; for (step-size / σ_{noise}) ≥ 2.5, it has subpixel position localization ($\sigma_{position} < 1/3$) and an angular localization better than 10°; further, it is insensitive to gradients. These results are demonstrated with analysis, statistical data and edgel-images. Also included is a comparison, of performance on a real image, with a typical operator (Difference-of-Gaussians). The results indicate that the proposed operator is superior with respect to detection, localization and resolution.

I. Introduction

An edge in an image corresponds to an intensity discontinuity in the scene. Although it may correspond to an edge of an object in space, it need not. It might well be the image of a shadow (illumination discontinuity) or a surface mark (reflectance discontinuity).

It is hard to over-emphasize the importance of edge-detection in image understanding. Most modules in a conceivable vision system depend, directly or indirectly, on the performance of the edge-detector. Consequently, there has been a substantial effort in this direction. Despite this effort, many in the community believe that the problem is largely unsolved. In fact, it may be claimed with some justification, that research and motivation on other fronts (e.g. stereo and line-drawing interpretation) has been dampened by the ineffectiveness of existing detectors.

Blicher [4] provides an insightful review of previous work on finding edges in image data [see also 7, 1]. Much of this work has been based on discrete approximation to differential operators [see 5]. Although edges do contain large first derivatives and zero-crossings of the second, the mapping is neither one-one, nor onto. It is well known that derivatives emphasize high-frequency noise. In fact, the higher the order of the derivative, the more pronounced the effect (taking the n^{th} derivative of a function is equivalent to multiplying its Fourier Transform by f^n). Further, operators that threshold on the first derivative respond to smooth shading. For example, the Nevatia-Babu Operator [13] and Canny's Operator [6] return false edges on smoothly shaded surfaces. Lateral inhibition has been proposed as a solution by Marr-Hildreth [11] and Binford [3]. However, this may involve taking 3^{rd} order derivatives.

The noise-characteristics of an operator depend on its size. The larger the operator, the more it averages out random noise. However, it is also more likely to overlap several edges or corners simultaneously and thus degrade the resolution capability. The detectability and localization of high-curvature edges also suffers. Further, as the operator size is increased, the assumptions invoked in its design may breakdown, introducing large and unknown biases. Whereas the noise sensitivity of an operator depends on its size, the associated resolution capability depends on the support used to make decisions. This support is generally larger than the operator-size. For example, one may use a (3 x 3) window to estimate the gradient at a point and then base the decision on a local gradient maximum whose detection requires considering at least three adjacent estimates. The lateral decision-support which determines resolution in this case, is 5

This is an updated version of a paper with the same title [12] which was published in the Proceedings of the Image Understanding Workshop held at New Orleans on October 3-4, 1984. It has been accepted for publication in the IEEE transactions on Pattern Analysis and Machine Intelligence.

This work was supported in part by the Defense Advanced Research Projects Agency under contract N00039-84-C-0211. During its early phase, V.S.N. was supported by the Information Systems Laboratory at Stanford.

pixels and not 3. Directional operators, like those of Nevatia-Babu [13] and Binford [3], introduce implicit averaging which is largely along the edge rather than across it. Isotropic operators, like Marr-Hildreth [11] and Shanmugam-Dickey-Green [17], on the other hand offer simplicity and uniformity at the expense of smoothing across edges. Gaussian smoothing has been employed by Marr-Hildreth [11] and Canny [6] to reduce noise. This can be decomposed into two orthogonal 1-D gaussian smoothing operations: one along the edge and the other across it. Let us consider the component along the edge. It is our claim that for a given support along a locally straight edge, gaussian smoothing is less effective than simple averaging. The argument runs as follows. Given N equal intensities, each with identically, independently distributed additive gaussian noise: the standard deviation of the weighted average of the intensities is minimized when the weights are all identical (the standard deviation in this case is reduced by a factor of $\sqrt{N}$). This argument can be equivalently carried over to the Fourier Domain. Gibb's phenomenon, although present, is not of any significance along the edge (it might, however, play a role at terminations).

Surface-fitting is among the other methods used to detect edges. It has been employed, both as a means to estimate derivatives, as by Prewitt [16] and Haralick [8], and as a classification technique, as by Hueckel [9]. The chief problem has been the choice of an adequate basis i.e. a basis which can accurately represent the feature sought to be detected. Further, these attempts have largely failed to exploit directionality.

Most of the previous work has ignored the "blurring" effect of the imaging system, which can be modeled approximately by gaussian convolution [see 2]. Blurring avoids undersampling, and aids, as will be shown later, facilitates sub-pixel localization of the edge.

Few [e.g. 9] have treated edges as composed of edgels, i.e. linear edge-elements, rather than edge-pixels. This directional information, we claim, is not only an essential descriptor of an edge at any point along its length, but is also valuable for linking [see 18].

Time and again, claims to "optimality" have been made. Among others, the claimants include Hueckel [9], Shanmugam-Dickey-Green [17] and Canny [6]. However, often the analysis is in the continuous domain, the assumptions and criteria questionable, and the extensions to 2-D ad hoc. An "optimal" solution is only as good as the optimality condition used.

In this paper, a variant of the surface-fitting approach is used; however, there are significant differences from most previous approaches. 1) An oriented *one-dimensional* surface, i.e. a surface constrained to be constant along some direction, is used. This results in effective noise-reduction without blurring the edges as severely as in circularly symmetric smoothing operators. 2) We do not seek to mark pixels as belonging to an edge, but to detect edgels, i.e. short, linear edge-elements, each characterized by a direction and position. 3) The blurring function of the imaging system, which is approximately gaussian, is taken into account. This results in sub-pixel localization of the edge. Sub-pixel localization could also be achieved by deconvolution followed by the localization of discontinuities. Deconvolution, however, is an ill-conditioned problem [see 2]. 4) An adequate basis has been found not only for most step-edges, but also for roof-edges and line-edges. These are various combinations of the tanh function with a constant. 5) Binford suggested [3] that it is desirable to do away with thresholds altogether. Any method which selects a subset of candidate edgels has implicit thresholds (in our case this corresponds to the selection of the best-fitting surface). However, the choice of an explicit threshold does not play a pivotal role in our scheme as in most others. This will be illustrated in Section VII. In fact, it may be desirable to postpone explicit thresholding to the linking stage.

We begin, in section II, by giving a definition of an edge in terms of the intensity profile of the viewed scene. Then, in section III, some of the problems associated with edge-detection based on zero-crossings of the second derivative are discussed. Much of the work to date has used a variant of this criterion. Section IV contains the details of our approach and Section V outlines the algorithm step-by-step as it has been implemented for step-edgel detection. This is followed in Section VI by a detailed example.

The proposed approach to edgel-detection is robust with respect to noise. For $(\textit{step size} / \sigma_{noise}) \geq 2.5$, it has sub-pixel position localization ($\sigma_{position} < 1/3$) and an angular localization better than 10°. Statistical supporting evidence in Section VII is accompanied by some simple analysis in Appendix III. Further, our operator is insensitive to high intensity gradients which do not correspond to edges. These claims are substantiated in section VIII, with pictures of edgels estimated from several images. This section also includes a comparison between the performance of an implementation of the Marr-Hildreth Operator [11] and our operator. The pictures presented, indicate that our operator is superior with respect to detection, localization and resolution. We conclude in section IX.

It should be pointed out that the problems of multiple scale and of linking edgels into extended edges are not considered here. Reliable edgel-detection should be expected to make these problems more manageable. Results on linked edges will be forthcoming in a sequel.

II. Definition of an Edge

Any extended edge in an image can be approximated by short linear segments called edgels, each characterized by a position and an angle. Edgels correspond to local discontinuities of various order in the intensity surface of a scene. A discontinuity of the n^{th} order is one whose n^{th} derivative contains a delta function. Hence, a line-edge is a 0^{th} order

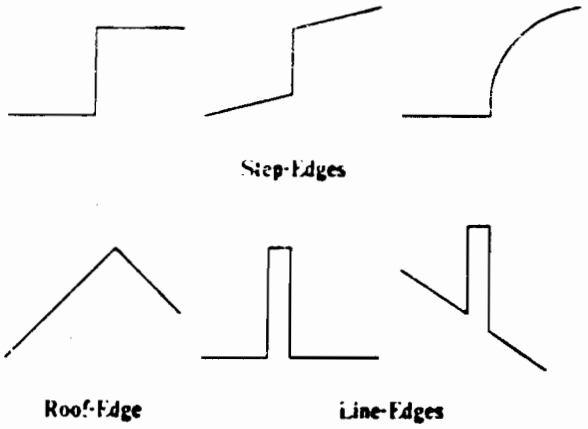


Fig 1. Examples of edge-profiles, as they appear before being "blurred" by the imaging system.

discontinuity, a step-edge is a 1^{st} order discontinuity and a roof-edge is one of 2^{nd} order. Some examples are shown in Fig. 1. However, the images we obtain in practice are degraded by optical and other aberrations. These effects can be approximated by convolution with a gaussian [see 2] of a certain standard deviation. Some of this "blurring" is desirable, even though it limits the resolution, because it also bandlimits the signal before it is sampled. Its absence would result in severe aliasing. A manifestation of aliasing in a picture would be the "staircase" appearance of edges which are neither horizontal nor vertical. (Beware of mistaking scanner-line jitter [see 2] for aliasing!) As a consequence of "blurring", there are no intensity-discontinuities in the image. The importance of this will be illustrated in the following sections.

III. Zero-Crossings of the Second Derivative

Much of the work to date has used zero-crossings of the second derivative to detect and/or localize step-edges [11, 8, 6 etc.]. There are some problems associated with this. As indicated in the introduction, derivatives amplify high-frequency noise. Further, if surface-fitting is used to estimate derivatives and the basis is inadequate, then the zero-crossing can result in extremely bad localization, e.g. consider the case of a cubic-fit for a step-edge cross-section which is located near the boundary of an image window (see Fig. 2).

It is not hard to see that we can have zero-crossings in the absence of an edge, e.g. at the base of a ramp [3]. Zero-crossings of the second-derivative are essentially points of inflection and these need not correspond to edges, as in the case of a corrugated intensity surface. It is our claim that zero-crossing operators do not adequately exploit the local intensity-profile of step edges.

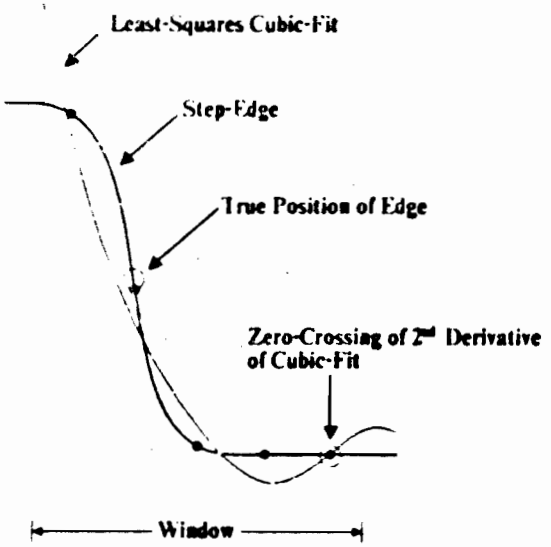


Fig 2. Inadequacy of the cubic-fit for a step-edge cross-section which is positioned near the edge of a window.

The intensity surface on the two sides of a step-edge will in general be sloped, as indicated in Fig. 1. We will henceforth refer to such an edge as a generalized step-edge. In contrast, an ideal step-edge is constant on both sides and is a subset of the former. Note that whenever we refer to an ideal or generalized step-edge in an image, the imaging-system "blur" will be implicit. A simple analysis (Appendix I) of a generalized step convolved with a gaussian shows that, in the continuous case, the localization based on zero-crossings would be biased by $(\Delta_{slope} \cdot \sigma^2_{blur} / \text{step-size})$, where Δ_{slope} is the difference between the slopes on the two sides of the step and σ_{blur} is the standard-deviation of the effective blurring gaussian mentioned in the previous section. On more than one occasion, authors have suggested gaussian preconvolution as a method of noise reduction [11, 6]. It can be shown that this would effectively amount to having a blurring function with a variance equal to the sum of the two variances and hence, it would further degrade localization of generalized step-edges.

IV. The Details

A variant of the surface-fitting approach is used here. However, unlike previous work, our basis is constrained to be directional and is non-linear in its parameters. It is important for the reader to distinguish between a non-linear basis and a basis which is non-linear in its parameters: to illustrate, $(a_0 + a_1 x + a_2 x^2 + a_3 y^3)$ is linear in its parameters while $(b + b^2 x)$ is not. Whereas any least-squares surface-fitting method whose basis is linear in

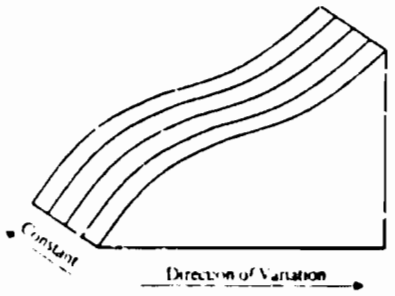


Fig 3. *One Dimensional* Surface

its parameters can equivalently be formulated as a convolution, surface-fitting with a basis which is non-linear in its parameters cannot be thus formulated.

We take into account the fact that the image consists of samples of the true intensity profile blurred by the imaging system. The standard deviation of this gaussian blurring function can be determined by an examination of the image of a point or step-edge. As a result of this blur, we have an image with no underlying discontinuities. The spectrum is bandlimited, avoiding aliasing and making sub-pixel localization possible.

The noise is generally assumed to be additive white gaussian. If one could find the direction of the edge in a reliable fashion, then the noise could be reduced by averaging data in a direction parallel to the edge. This, of course, relies on the fact that the window, i.e. the kernel of the operator, is small enough for the edge-segment in it to be modeled as an edge! We achieve the above mentioned smoothing by fitting to each window a *one-dimensional* oriented surface, i.e. a surface which is constant in one direction, as shown in Fig. 3 (the direction of invariance would be parallel to the edge!). Fitting this *1-D* surface is equivalent to treating the data as strictly one dimensional by projecting it along the direction of invariance onto a plane.

Now we come to the question of a reliable direction-finder for windows hypothesized to contain edges. A first-approximation for the direction of variation can be obtained from the gradient of a least-square-error planar-fit to the window. However, this leads to a substantial systematic bias for rectangular windows [11], which is what we have used. A more general surface can be used to refine this first estimate and reduce the biasing error. We fit a least-squares *one-dimensional* cubic surface to the nearest 5^2. To clarify, a *1-D* cubic surface is constant in one direction and is described by a cubic polynomial in the orthogonal direction. Starting with the initial estimate of the direction, which is obtained from the planar-fit, the search for the orientation of the cubic-fit is generally not more than a few steps. It should also be pointed out that for a window with an edge!, the plot of the square-error vs angle for a *1-D* cubic-fit is bowl-shaped and centered around the true angle. Hence, once within the bowl, standard techniques like Newton's Method can be used to find the minimum. Appendix II contains all the relevant equations for the various least-square-fits performed.

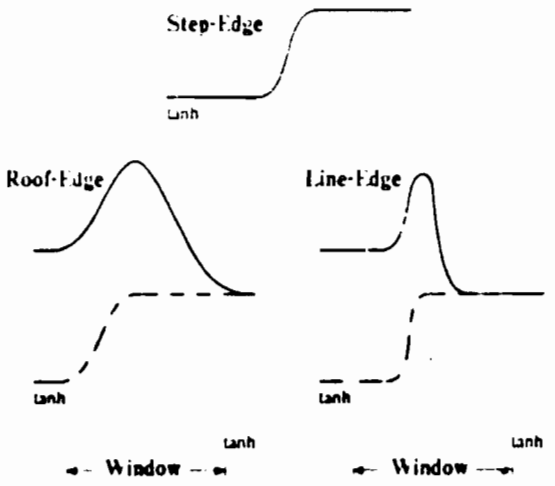


Fig 4. *Adequate* bases for edge-profiles in the image are combinations of the tanh function with a constant.

It should be emphasized that there cannot be any one unique basis which is appropriate to describe the image data in all windows. If we attempt to do this, we will obtain incorrect results when the basis is inadequate and noise-sensitive results if the basis is not minimal. Perhaps, a simple illustration of this important observation is called for. Consider a hypothetical situation where we are given some noisy samples of "y," which is a polynomial in "x," and are told to determine a description of the underlying curve. For the sake of argument, assume that "y" is a quadratic function of "x." Obviously, fitting a straight line ($y = a_0 + a_1 x$) to the data is not going to give us an adequate description. More importantly, fitting a polynomial in "x" which is of higher order than a quadratic, is going to have non-zero coefficients for x^i, $i > 2$, owing to the noisy nature of the data. Hence, even though polynomials of order greater than 2, give a smaller least-squares-error than a polynomial of order 2, it is desirable to fit a quadratic. The reader will probably recognize this to be a restatement of the underlying principle of linear regression analysis in statistics. Considerations similar to the ones just detailed have been investigated for one dimensional steps by Leclerc and Zucker [10].

Now, consider the choice of an adequate basis. For most step-edges the tanh function with a constant, i.e. $a\tanh(f[z+p]) + k$ where a, p and k are the parameters and f is a constant determined by the "blur" of the imaging-system will be adequate.

As can be seen from Fig. 5, the maximum error in approximating an ideal step-edge by the tanh is less than 1% of the step-size. One important by-product of employing the tanh is a reliable estimate of the contrast of the edge. From our case studies it seems that the contrast is helpful not only in linking, but also in interpretation. For roof-edges and line-edges, combinations of the tanh function, as depicted in Fig. 4, seem to be adequate bases.

Some authors have tried to detect edges which have large deviations from an ideal step-edge by using multiple scales. Multiple scales, we believe, are unnecessary and undesirable in this case. For such edges, the tanh basis is inadequate and a cubic or a tanh with a cubic might be adequate. The latter has some problems because the tanh and cubic are not completely independent. It should also be noted that the cubic is inadequate for most step edges and that derivative estimates based on a cubic-fit can be quite unreliable due to the *wiggles* which are characteristic of polynomials. It may be desirable to employ splines when the tanh and the cubic are inadequate bases. We have used a cubic, with a check for consistency in position estimate with the tanh-fit, in one version our detector. Our window is too small (5 x 5) for finding the parameters of a tanh with a cubic or of splines, in the case of horizontal and vertical edges. The position estimate based on the zero-crossing of the second derivative for a cubic-fit is biased for reasons similar to those listed in Appendix I. Hence, for large values of Δ_{slope}, refinement of the initial estimate may be desirable. If one uses a general basis like the cubic, it is also desirable to confirm that a dominant component of the cubic-fit is indeed a step-edge. In our implementation we accomplish this by basing our estimate of the step-size on a tanh-fit even if the basis used for detection is the cubic. We do not consider our handling of non-ideal steps to be completely satisfactory.

We compare the least-square-error of a quadratic-fit with that of a tanh-fit and choose the one with the smaller error to determine the existence or absence of a step-edge[4]. This discriminates against smooth shading and reduces the significance of subsequent thresholding. In the initial stages, we had used the χ^2-Statistic to determine the adequacy of the basis. It was found that this was unnecessary and perhaps undesirable because of inadequate modeling of the error. A procedure similar to the one just described can be used to detect steps with large deviations from an ideal step-edge. For example, if a cubic basis is being used, then the 1, 20 F-Statistic corresponding to the quadratic and cubic fits should be employed to verify the appropriateness of the cubic-fit.

[4]It should be noted that both the fits have the same number of unknown parameters. This justifies our comparison of the two least-square-errors to determine which basis describes the data more accurately. The formulation of the F-Statistic corresponding to the tanh and quadratic fits is not possible, even if one ignores the non-linearity, because the bases are not nested.

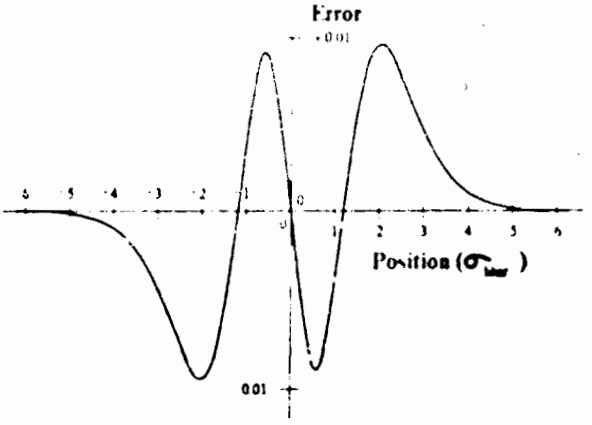


Fig 5. Error profile resulting from the approximation of an ideal unit step-edge by $\left[0.5 + 0.5 \tanh\left(\frac{0.85\ x}{\sigma_{blur}}\right)\right]$.

At this juncture, we would like to bring to the reader's notice some of the reasons to expect an improved performance from the use of a directional tanh-surface. First, our basis requires the specification of only four parameters which determine the orientation, the position and the upper and lower intensities of the step-edge. It is immediately seen that this is the minimum number required to describe a step-edge with predetermined 'blur'. Contrast this with eight required by Heuckel's Method [9] and ten by Haralick [8]. As a result, we can use smaller windows than most previous equally sophisticated approaches. This implies better resolution capabilities and improved performance on high-curvature edges. Second, the highly constrained nature of our basis (which is borne out by the presence of only four unknown parameters) should be expected to offer noise-robustness analogous to matched-filtering classification wherein noisy patterns are categorized based on their closest "match" to noiseless representatives of the different classes. Our approach distinguishes between two classes: step-edges and non-step-edges. Step-edges are characterized by a step-component of variable intensity, orientation and position. Non-step-edges can be better described by quadratic surfaces. Of course, these assumptions may break down as the window size is increased.

We have carried out our initial investigation for step-edges, which are by far the dominant type. Numerically, it was determined that for $\sigma_{blur} = 1$ and an ideal step-edge, the optimum scaling factor for the argument of the tanh function was 0.85. This factor was determined by minimizing the square-error. This is not surprising, as equating the slopes of the two functions at the origin would give us a value of 0.8. Hence, a rule of thumb for the scaling factor is $(0.85/\sigma_{blur})$. The normalized error-profile, using this factor, is shown in Fig. 5. The detection scheme is

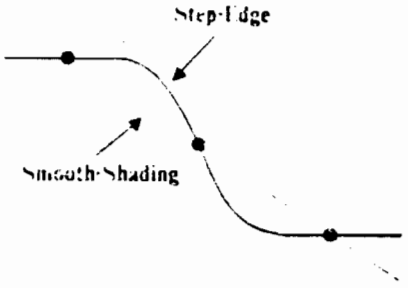


Fig 6. Ambiguity in profile, given 3 symmetric samples of a step-edge cross-section.

not particularly sensitive to this factor and, in fact, it detects reasonably diffuse shadows.

The window size is determined by the standard deviation of the blurring gaussian. It is not hard to see that the minimum window size, irrespective of the "blur", has to be larger than (3 x 3) because, as illustrated in Fig. 6, there is no way to distinguish a horizontal or vertical step-edge from smooth shading if we take three symmetric samples of the edge. We chose (5 x 5) square windows. Not surprisingly, detection of zero-crossings of the 2nd derivative requires a minimum lateral support of 5 pixels i. the symmetric case. As the window size is increased for a fixed blur, we trade-off resolution for improved detection and localization of locally straight edges. However, the detection and localization of high-curvature edges will deteriorate because of the invalidity of our implicit edgel-model. (Resolution refers to the minimum support required for the detection of an edgel, i.e. an edgel is theoretically resolvable if it can be isolated within any window. If it is not detected, it is be due to the inadequacy of the edgel-detector. For example, if three parallel edges are spaced at 2-pixel intervals, then with our choice of support, the middle edge would not be resolvable, but the other two might be. We will point out examples of these in our first case study. It should be noted that we have not investigated the tradeoffs accompanying different window shapes.

V. Outline of Algorithm for Step-Edgel Detection

The following is an outline of the procedure used to detect the presence of an edgel in an image-window. This procedure is to be repeated over the whole image by shifting the window in 1-pixel steps in the x and y directions.

(All the relevant equations and statistics are listed in Appendix II.)

(i) Perform a least-squares planar-fit to the window and use the gradient of this fit to estimate the direction of variation in the window, assuming that the underlying intensity surface is *1-D*.

(ii) Refine the estimate of the direction of variation by fitting a *1-D* cubic surface with the least-squares-error criterion. The resulting equations are non-linear in the angle. However, owing to the reliable initial estimate, the search is typically a couple of steps. We find the angle to the nearest 5°.

(ii) *{Optional}* Calculate the 2, 20 F-Statistic for the planar and cubic fits obtained in (i) and (ii). If it is less than the 75% threshold, then declare the absence of an edgel. This thresholding serves the purpose of reducing computation by considering only those windows which exhibit a statistically significant reduction in the least-square-error by employing a cubic basis rather than a planar one.

(iii) Find the least-squares *1-D* tanh surface oriented in the direction found in (ii). The tanh-fit is localized to the nearest 0.1 pixel. As will be seen in Section VII for low and moderate S/N R's the position accuracy is not determined by the quantization error associated with the search steps.

(iv) Find the least-squares *1-D* quadratic surface oriented in the direction found in (ii). If the least-squareserror in this case is less than that for the tanh-fit, then declare the absence of an edgel.

(v) The least-squares tanh-fit performed in (iii) determines the intensities on the two sides of the step and its position in the window. The sum and difference of the constant term in the basis and the coefficient of the tanh term determines the intensities and the position of the step-edge is given by the displacement of the tanh term.

(vi) Threshold on the step-size determined from (v). To improve the reliability of the detection process, it may also be desirable to require the edgel to be localized within some central sub-window, e.g. 2-pixel X 2-pixel.

NB If one wants to detect step-edgels which have large deviations from an ideal step-edgel, steps similar to (iii) and (iv), but with a basis different from the tanh, will have to be added. Of course, the appropriate statistical formulation will also have to be used.

VI. An Example

We now proceed to illustrate the algorithm outlined in the previous section with an example. Consider the image-window in Fig. 7-b which is a noisy version of that in Fig. 7-a. The underlying intensity step-edge shown in Fig. 7-a has grey-levels 64 and 128 on its two sides and $\sigma_{blur} = 0.6$. The edge is located at a distance of 0.2363 pixel from the center of the window and at an angle of 31.7° to the x-axis. The noise in Fig. 7-b is additive white gaussian with $\sigma_{noise} = 8$. The detected edge is located at a distance of 0.1679 pixel from the center of the window

Underlying Edge

128	128	128	124	108
128	127	118	97	75
124	110	86	69	64
99	76	66	64	64
70	64	64	64	64

Fig 7-a. Example : Original image-window with step-size = 64.

Detected Edge

135	132	132	127	118
121	133	110	101	88
111	110	72	75	61
108	78	61	69	71
71	63	53	61	66

Fig 7-b. Example : Noisy image-window with (step-size / σ_{noise}) = 8.

and at an angle of 30° to the x-axis. The error in position is 0.0684 pixel and the error in angle is 4.4°. Recall that the position quantization error is ± 0.05 pixel and that the angle quantization error is ±2.5°. As mentioned in the previous section, the relevant equations are listed in Appendix II. The z-axis shown in Fig. 7-c is the estimated direction of variation in the window and is orthogonal to the estimated orientation of the edge.

(i) Least-Squares Planar-Fit

$I[x, y] = 74.72 - 7.34x + 16.52y$
Least Squares Error = 2683

$\theta_0 = \tan^{-1}\left[\frac{16.52}{-7.34}\right] = 115°$ (*to nearest* 5°)

θ_0 is the direction of the gradient of the planar-fit and is used as a first estimate for the direction of variance in the window.

(ii) Least-Squares 1 *D* Cubic-Fit

$I[x, y] = 71.74 + 21.83z + 6.19z^2 - 2.16z^3$
$z = x.\cos(\theta) + y.\sin(\theta)$, $\theta = 120°$ (*to nearest* 5°)
Least Squares Error = 1295

θ a refined estimate of θ_0, is the final estimate of the direction of variation in the window and is orthogonal to the direction estimate for the edgel, if any.

(ii)' {*Optional*}

The 2, 20 F-Statistic for the planar and cubic fits is 10.7 and it does exceed the 75% threshold which is 1.47. Hence, we continue with the rest of the algorithm.

(iii) Least-Squares 1 *D* Tanh-Fit along θ

$I[x, y] = 95.57 + 32.52\tanh\left[\left[\frac{0.85}{0.6}\right].(z - p)\right]$

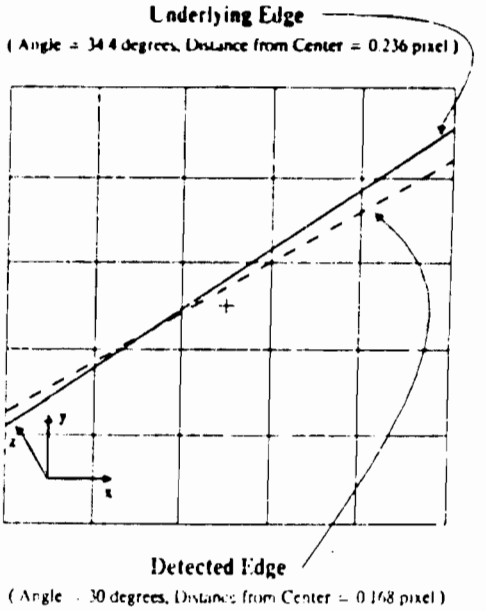


Fig 7-c. Example : Underlying and detected edges.

$p = 0.9$ (*to nearest* 0.1 *pixel*)
$z = x.\cos(120°) + y.\sin(120°)$
Least Squares Error = 1203

p is the estimate of the position of the edgel along the z-axis.

(iv) Least-Squares 1-D Quadratic-Fit along b

$I(x, y) = 77.80 + 15.86z + 1.45z^2$
$z = x\cos(120^\circ) + y\sin(120^\circ)$
Least Squares Error = 2615

The quadratic-fit least-squares-error is more than the tanh-fit least-squares-error. Hence, an edgel has been detected.

(v) Edge Parameters

The intensities on the two sides of the step are estimated from (iii) to be 63.1 and 128.1 (i.e. 95.57 ± 32.52). The orientation of the edgel is determined from (ii) to be 30°, i.e. orthogonal to θ, the direction of variation. Its position is determined from (iii) to be 0.9 pixel from the origin along the z-axis or equivalently 0.1679 pixel from the center of the window.

VII. Statistical Data

We now present some statistical results obtained from our edges-detector. The algorithm outlined in steps (i) through (vi), excluding (ii)', of Section V, was implemented. Let us begin by clarifying our notation. Signal-to-Noise-Ratio (S.N.R.) is defined as $(step\ size / \sigma_{noise})$, where σ_{noise} is the standard deviation of the noise. The noise is assumed to be additive white gaussian. A false-positive occurs when no edge is present in the window and an edge with contrast greater than the threshold is declared. A true positive occurs when an edge is present in the window and it is identified as such, with its contrast greater than the threshold, the error in position (perpendicular distance from the center of the window) less than 0.7 pixels (half the diagonal of a pixel-support) and the error in angle less than 15°. $\sigma_{position}$ is the root-mean-square of the error in the position and σ_{angle} is the root-mean-square of the error in the angle. (We use σ to denote the r.m.s. values because they closely approximate the standard deviation of the errors. This is a consequence of the bias in the position and angle estimates being relatively small. The threshold is on the edge-contrast and is always stated in units of σ_{noise}.

Fig. 8 shows a plot of false positives vs threshold. Windows of size (5 x 5) with a constant intensity surface, $\sigma_{blur} = 0.6$ and additive white gaussian noise were used for this simulation. The value of σ_{blur} was chosen to be 0.6 because this was found to be its estimate in the real images considered in the next section. It can't be much smaller than 0.5 as then we should expect aliasing and if it's much larger, the edge is at a larger scale and we need a correspondingly larger support. Notice, that even for a zero threshold, false positives are declared in only 31% of the cases. This is in contrast with gradient thresholding schemes which would give 100% false positives. That is because our detection scheme requires a certain step-like "correlation" among the samples for an edge to be declared. This requirement stems from our choice of the tanh as a basis. We have F.P. < 2.5% for a threshold of $1.5\sigma_{noise}$; F.P. < 0.2% for $2\sigma_{noise}$ and F.P. < 0.01% for $2.5\sigma_{noise}$.

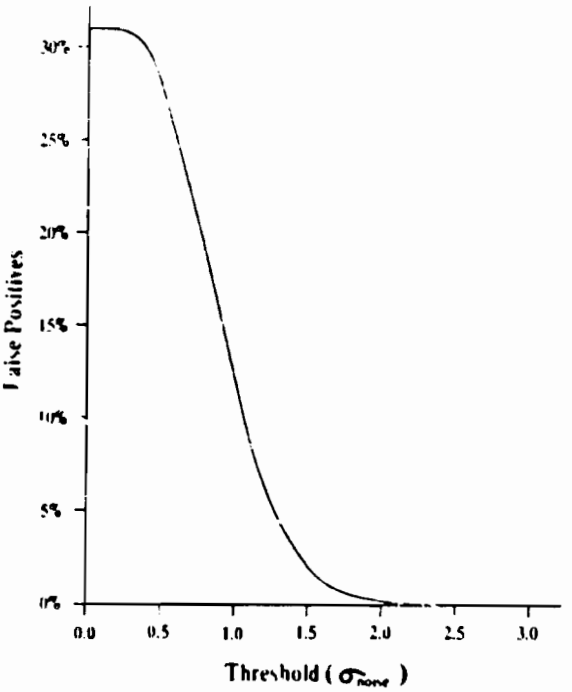


Fig 8. Plot of false positives detected in windows with constant intensity and additive white gaussian noise, as a function of the threshold.

Fig. 9 shows a plot of the true positives vs. the threshold. Square (5 x 5) windows with ideal step-edges, $\sigma_{blur} = 0.6$ and additive white gaussian noise were used for the simulation. Each step-edge passed through a 1-pixel square in the center of the window and its position and angle were independently uniformly distributed. Constraining the edge to pass through the central pixel-support is justified because each segment of an edge, which is not near the picture border, will pass through the center-pixel of one window or another. To reduce the contribution of gray scale quantization effects, the edge contrast was chosen to be 64 levels on a scale of 0-255. Notice, that even for zero threshold, we do not get 100% detection for low S.N.R.'s. In contrast, gradient thresholding schemes would give 100% true positives. But then, they would declare any distribution to be an edge! Thus they would have 100% false positives too. Also notice the relatively flat profile of the plots when the S.N.R. is less than the corresponding threshold (the "knee" of the plot for a particular S.N.R. occurs when the threshold is equal to the step-size). If we synthesized images rather than windows, we should expect somewhat higher detection since each non-border edge-segment in a pixel-support is "scanned" 25 times and as pointed out earlier, we detect edgels and not edge-points.

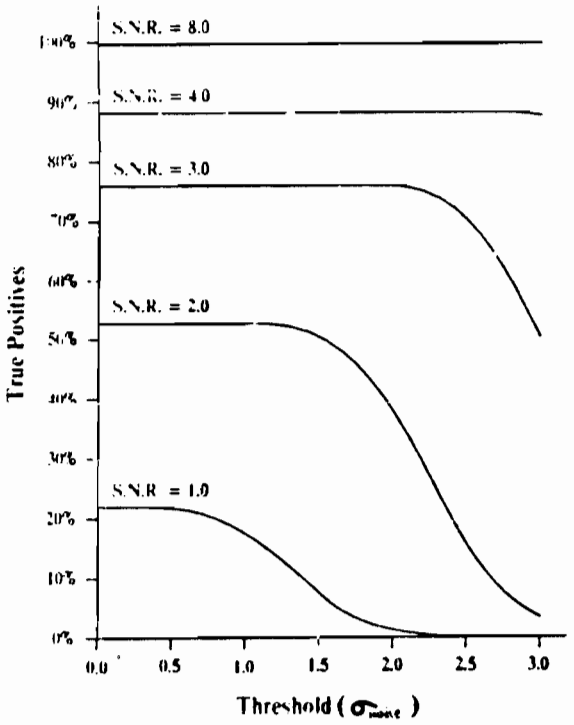


Fig 9. Plot of true positives detected in windows with synthesized step-edges and additive white gaussian noise, as a function of the threshold for different S.N.R.'s.

Fig. 10 shows the plot of σ_{angle} vs S.N.R. for the true positives which would be detected in Fig. 9 if the threshold were zero and the constraints in the position-error and angle-error were removed. This curve decays to $\sigma_{angle} \approx 1.9°$ for large S.N.R.'s (the diagram is cut-off at S.N.R.=8). This is about 30% more than what we should expect from the quantization error for a uniformly distributed random variable [15]. It suggests that the bias associated with cubic-fit angle estimates, from the (5 x 5) windows, is small in comparison to the quantization interval, i.e. $^{\circ}$

Fig. 11 shows the plot of $\sigma_{position}$ vs S.N.R. under the same conditions as in Fig. 10. This curve decays to $\sigma_{position} \approx 0.032$ for large S.N.R.'s (the diagram is cut-off at S.N.R.=8). This differs by about 10% from what we expect from the quantization error for a uniformly distributed random variable. This suggests that the position estimates from the (5 x 5) windows have a negligible bias in comparison to the quantization interval. In Appendix III, we derive an expression for $\sigma_{position}$ for the 1-D high-S.N.R. case. It can be shown that this is equivalent to a vertical or horizontal edge in the present simulation, with the effective S.N.R. being $\sqrt{5}$ times the actual S.N.R.. It turns out that the values we would expect for vertical or horizontal edges using the expression derived in the Appendix are are within 25% the those shown in Fig. 11. This is despite the fact that the errors in the

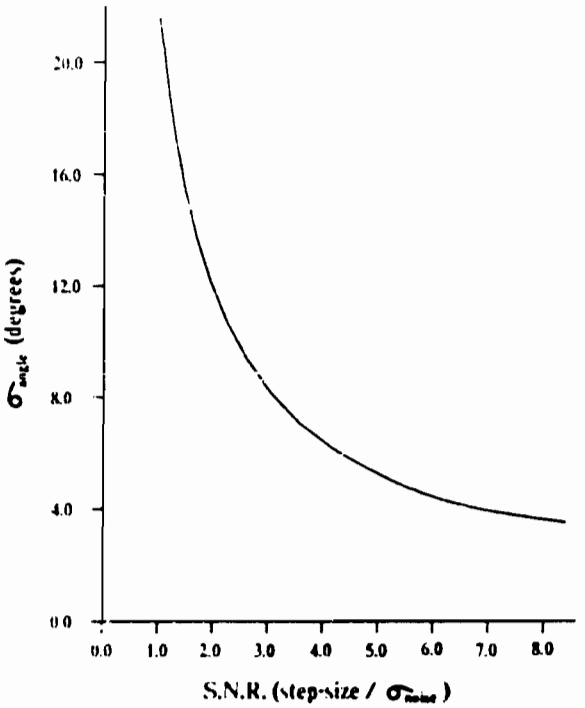


Fig 10. Plot of the standard deviation of the angular error in the true positives detected with threshold = 0, vs the S.N.R..

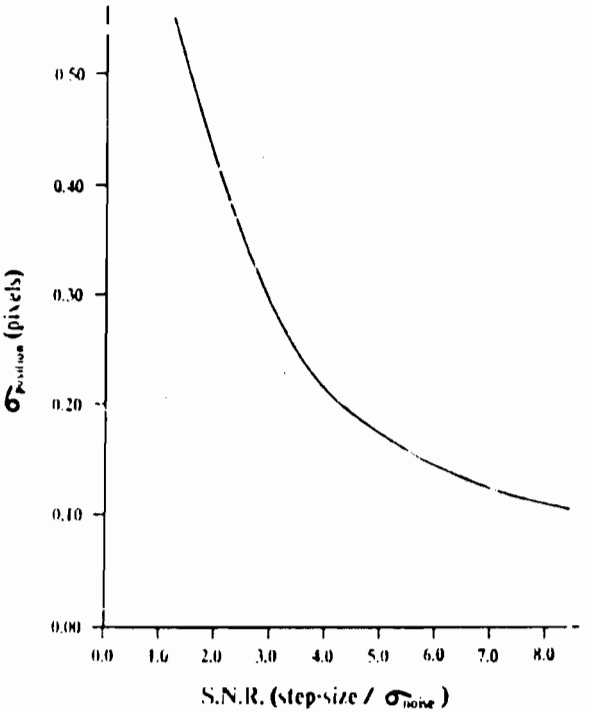


Fig 11. Plot of the standard deviation of the positional error in the true positives detected with threshold = 0, vs the S.N.R.

angle estimates propagate to introduce errors in the position estimate. The asymptotic value is within 10%. This confirms the domination of the quantization error for high S.N.R.'s.

The reader may also wish to know the effect of the inclusion of step (ii)' on the statistics. Although the shapes of the false-positives and true-positives plots remain more or less the same, their sizes get scaled. The false-positives plot now starts out at 11% for a zero threshold and decays to F.P.< 1% for a threshold of $1.5\sigma_{noise}$; F.P.< 0.1% for $2\sigma_{noise}$ and F.P.< 0.01% for $2.5\sigma_{noise}$. The plot of true-positives for S.N.R.=8 remains unchanged, the plot for S.N.R.=4 now starts out at 78% instead of 88%, S.N.R.=3 at 58% instead of 76%, S.N.R.=2 at 32% instead of 53% and S.N.R.=1 at 11% instead of 22%. The plots of σ_{angle} and $\sigma_{position}$ remain approximately the same.

Comparisons of the statistics of various operators are valid only if the the size of the support used to make decisions is the same. As indicated in the introduction, this need not necessarily be the same as the window size. Increasing the support size, which in our case is (5 x 5), would increase the fraction of true positives and decrease the fraction of false positives, for any given S.N.R.. Also, $\sigma_{position}$ and σ_{angle} would decrease. This, however, is at the expense of resolution between adjacent edges and the detection and localization of high-curvature edges.

We end this section with a word of caution. The analysis in the Appendix and the statistical data of this section are for ideal step-edges. They can at best, only be indicative of the performance on real images owing to the numerous simplifications and assumptions invoked. For example, non-constant intensity surfaces have a higher likelihood of false positives than constant surfaces like those used for the statistics. The results are of no value if our inherent edge-model is seriously flawed. Hence, although theoretical and statistical support is desirable, the proof of the pudding is in the eating.

VIII. Three Case-Studies and a Comparison

It is essential to point out some details concerning the photographs: a) Only the step-edgel detector output (including (ii)') has been implemented. (The step (ii)' marginally improves low contrast edges.) b) The edgels are displayed as squares of their edges proportional to the gradient magnitudes characteristics of the display device. Hence edges seem thicker than they are. This also occurs in the highlights. This can easily be compensated for. The superimposed images have been thresholded in all cases, for gradients greater than about $2.5\sigma_{noise}$. d) All edges displayed are computed from *raw* edgels with no post-processing, like linking, thinning, cleaning etc.. e) Degradation resulting from the various reproduction processes would make it difficult to confirm some of the edges present in the original image. This is particularly true in the high intensity regions which saturate the display well below the highest gray level. f) The pictures with the edgels and the superimposed edgels are displayed on a grid with twice the linear resolution of the original image because of our sub-pixel localization. Further, pixels in the vicinity of edgels have been reduced to the lowest gray level, for clarity. g) It is important to bear the size of the original image in mind when scrutinizing the pictures.

Fig 12-a. Bin of Parts : Original Image (128 x 128)

Fig 12-b. Bin of Parts : Edgel Image

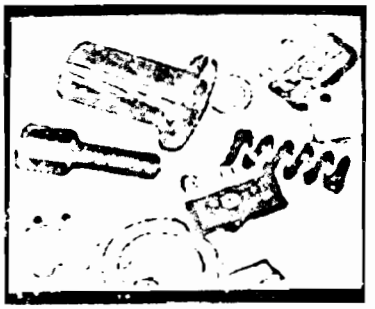


Fig 12-c. Bin of Parts : Superimposed Image

(i) Industrial Setting : Bin of Parts
(*Size* : 128 x 128; σ_{filter} : 0.6; σ_{noise} : 3)

Refer to Figs. 12-a (the original image), 12-b (the edgel image) and 12-c (the superimposed image). This picture was chosen to demonstrate the resolution capability of the detector and its perfor-

Fig 13-a. San Francisco Bay : Original Image (256 x 256)

Fig 13-b. San Francisco Bay : Edgel Image

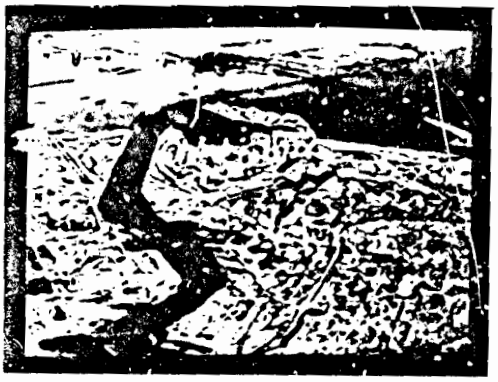


Fig 13-c. San Francisco Bay : Superimposed Image

mance on high-curvature edges. The pins of the various parts have false negatives. This is because they are bounded by dark lines, and our edge detector has currently been implemented only for step-edgels. The outer edges of the lines have been detected although not well-localized, but the inner edges exceed the resolution capabilities of our detector. Notice that some of the circular regions detected have a diameter of just a few pixels.

(ii) Aerial View : San Francisco Bay
(*Size* : 256 x 256; σ_{blur} : 0.6; σ_{noise} : 5)

Refer to Figs. 13-a (the original image), 13-b (the edgel image) and 13-c (the superimposed image). This picture was chosen because of its complexity. On first glance, it may seem that there are a host of false positives. However, a closer examination of the superimposed image reveals this to be untrue. The long lines in the sea correspond to silt lines. It may not be possible to confirm them in the photographs you will see. In any case, notice the continuity in most edges. Long continuous false positives are statistically unlikely. Also notice, the detection of the small island in the mid-right of the image. In the superimposed image, the edgels are seen to impose a structure based on local intensity changes.

Fig 14-a. Indoor Scene : Original Image (256 x 256)

(iii) Indoor Scene : Telephone, Cup and Pencil
(*Size* : 256 x 256; σ_{blur} : 0.6; σ_{noise} : 4)

Refer to Figs. 14-a (the original image), 14-b (the edgel image using the tanh-fit), 14-c (the edgel image using the tanh/cubic fit) and 14-d (the superimposed image). This image was chosen to illustrate the inadequacy of the tanh basis to deal with step-edges having a large non-zero slope on either side. Note the top surface of the telephone. It does not correspond to an ideal step-edge, but to a generalized step which some detectors might find by using a larger scale. The same is true of the top edge of the pencil. As can be seen from Fig. 14-b, these edges are

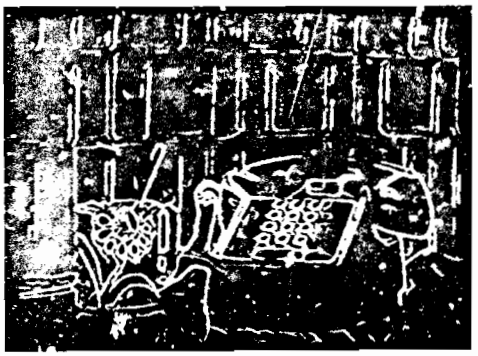


Fig 14-b. Indoor Scene : Edgel Image (tanh)

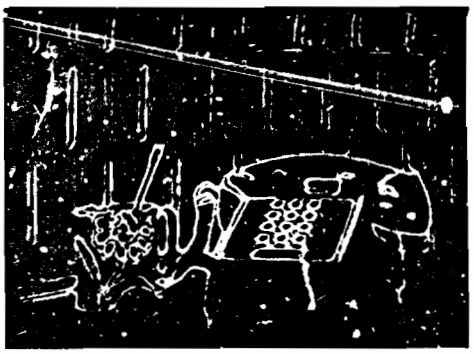


Fig 14-c. Indoor Scene : Edgel Image (tanh/cubic)

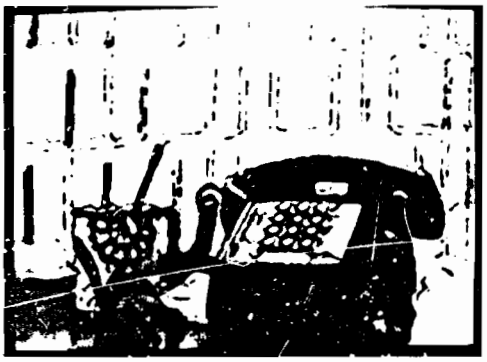


Fig 14-d. Indoor Scene : Superimposed Image (tanh/cubic)

missed if we use a tanh-fit at a single scale. This is a result of the inadequacy of the basis, i.e. the tanh and a constant cannot closely approximate step-edges which have large deviations from zero slope on either side. Using a tanh/cubic fit, as explained in Section IV, rectifies this without recourse to a different scale, as evident from Fig. 14-c. The only prominent edge which seems to have been missed is that of the table behind the book. This was due to its larger scale, which was confirmed by the examination of its profile at the individual pixel detail. The inner portion of the flower on the cup has a few false negatives due to lack of resolution. The superimposed image once again exhibits localization.

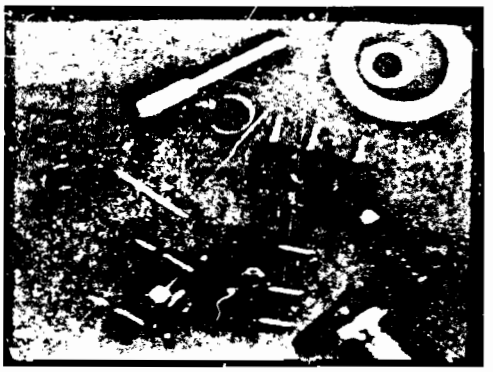


Fig 15-a. Original Image (256 x 256)

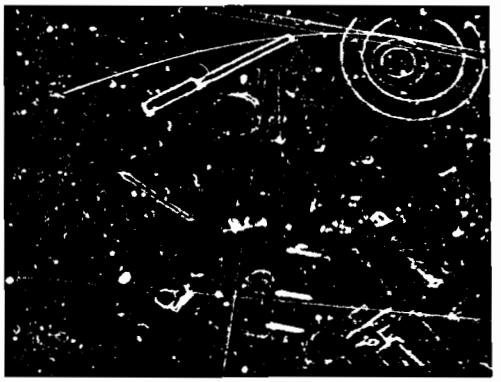


Fig 15-b. Edgel Image - Our Detector (tanh/cubic)

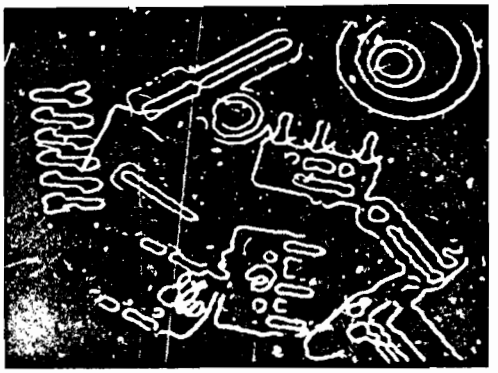


Fig 15-c. Edge Image - Marr-Hildreth Operator

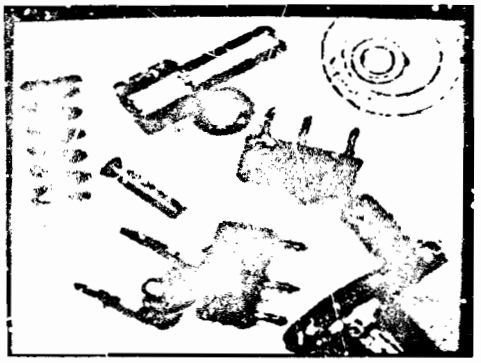


Fig 15-d. Superimposed Image - Our Detector (tanh/cubic)

Fig 15-e. Superimposed Image - Marr-Hildreth Operator

Fig 15-f. Close-Up of Superimposed Image - Our Detector (tanh/cubic)

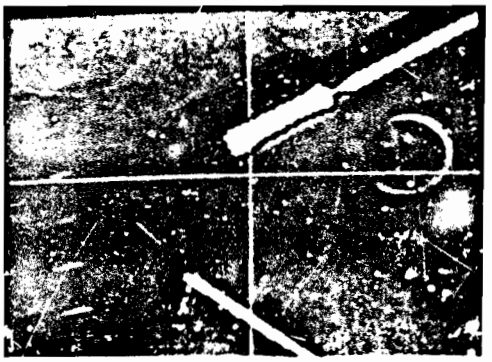


Fig 15-g. Close-Up of Superimposed Image - Marr-Hildreth Operator

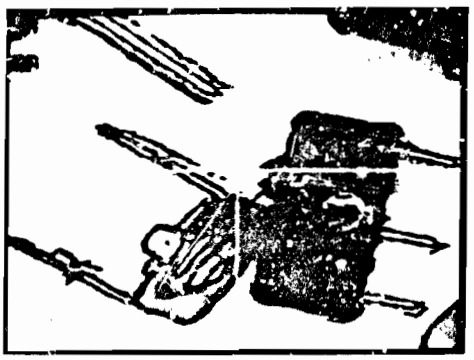


Fig 15-h. Close-Up of Superimposed Image - Our Detector (tanh/cubic)

Fig 15-i. Close-Up of Superimposed Image - Marr-Hildreth Operator

(iv) A Comparison : Bin of Parts
(*Size* : 256 x 256; σ_{blur} : 0.6; σ_{noise} : 2)

Refer to Figs. 15-a (the original image), 15-b (the edgel image for our detector), 15-c (the edge image for a version of the Marr-Hildreth Operator† [11]) and 15-d, -e (the corresponding superimposed images). Our detector used a tanh/cubic fit, as explained in section IV. In order to facilitate a comparison between the two

†The choice of the Marr-Hildreth Operator was based solely on convenience. It was used by S.R.I. International for the I.T.A. Project in which Stanford was also a participant. As the displayed image was among those used in the Project, we expect that the operator has been tuned for optimum performance on it. The implementation used the Difference of Gaussians (D.O.G.) with $\sigma_c = 1.6$, with a (11 x 11) support, and $\sigma_e = 1$, with a (7 x 7) support. The choice of $\sigma_c/\sigma_e = 1.6$ results in a close approximation to the laplacian of a gaussian [11]. The zero-crossings were thresholded on their slope. It is conceivable that a different implementation of the operator will produce better results, but it is unlikely that the improvement will be dramatic.

superimposed images, we *zoom-in* on (128 x 128) subsections in Figs. 15-f, -g, -h and -i. For reasons mentioned in the beginning of this section, it might not be possible to confirm all the detected edges. In any case, a careful examination is instructive to discover the differences in performance between the two operators with respect to detection, resolution and localization (especially of high curvature edges).

IX. Conclusion

This paper deals with the problem of edge-detection using directional *one-dimensional* surfaces. Edges were defined in terms of short, linear segments called edgels. Detection of edgels was claimed to be more appropriate than that of edge-pixels. Some shortcomings of derivative operators were then presented. An adequate basis for most step-edgels was shown to be the tanh. It is likely that other adequate bases exist and, in fact, if one were going to use a table look-up to perform surface fitting, the exact profile of the ideal step-edge can be stored. This is the integral of a gaussian and has no closed-form solution. A detailed discussion on the design of an operator was followed by an outline of the algorithm and an example. Robustness to noise, sub-pixel position localization ($\sigma_{position} < 1/3$) and better than 10^0 angular localization were statistically established for S.N.R. $\geq$ 2.5. This was accompanied by some simple analysis and a variety of images demonstrating the performance of our operator. In the course of the paper, it was indicated that our handling of non-ideal step-edges by using a cubic basis is not completely satisfactory.

An attempt was made to highlight some of the issues and concerns in edge-detection, as we see them. Analytical, statistical and empirical tools were employed to demonstrate the performance of the proposed algorithm. No attempt has been made yet at computational efficiency. We have concerned ourselves solely with accuracy. The current implementation is in Pascal on a VAX 11/780. The processing time is typically 20 C.P.U. minutes for a (128 x 128) image. We expect to reduce that by a factor greater than 2. The algorithm is implementable as a strictly parallel process and has natural extensions for roof and line edges.

Appendix I : Zero-Crossing Bias

Let $E(x)$ be a generalized step of height S at the origin, and $G(x)$ be a normalized gaussian with "standard deviation" σ_{blur}.

$$E(x) = \begin{cases} k_1 x & \text{if } x < 0 \\ k_2 x + S & \text{if } x > 0 \end{cases}$$

$$G(x) = n.e^{-x^2/2\sigma_{blur}^2}$$

Then $(E(x) * G(x))$ is the corresponding step-edge (where $*$ denotes convolution) and it can be shown, that $(E(x) * G(x))'' = E(x) * G''(x)$.

$$\begin{aligned} E(x) * G''(x) &= \int_x^{+\infty} k_1.(x-u).G''(u)\,du \\ &\quad + \int_{-\infty}^{x} [k_2.(x-u) + S].G''(u)\,du \\ &= \int_{-\infty}^{+\infty} k_1.(x-u).G''(u)\,du \\ &\quad + \int_{-\infty}^{x} [(k_2 - k_1).(x-u) + S].G''(u)\,du \\ &= [S - (k_2 - k_1).x].G'(x) \\ &\quad + (k_2 - k_1).[x.G'(x) - G(x)] \\ &= S.G'(x) - (k_2 - k_1).G(x) \end{aligned}$$

Equating this to zero we get, $x = \Delta_{slope}.\sigma_{blur}^2/S$ where $\Delta_{slope} = (k_1 - k_2)$ and S is the step-size. This is the biased zero-crossing of the second derivative.

Appendix II : Least-Squares Criteria & Statistics

Least-Squares Criterion for a Planar-fit :

$$\xi_P = \sum_{x,y=0}^{4} (Image[x,y] - (a_0 + a_x.x + a_y.y))^2$$

(minimize w.r.t. a_0, a_x and a_y)
Initial Estimate of θ : $\theta_0 = \tan^{-1}(a_y / a_x)$

Least-Squares Criterion for a Quadratic-fit :

$$\xi_Q = \sum_{x,y=0}^{4} (Image[x,y] - (a_0 + a_1.z + a_2.z^2))^2$$

(minimize w.r.t. a_0, a_1 and a_2)
$z = x.cos(\theta) + y.sin(\theta)$
θ is determined from the L.S.E. cubic-fit and is the angle by which the axes have to be rotated to align the x-axis with the edgel cross-section.

Least-Squares Criterion for a Cubic-fit :

$$\xi_C = \sum_{x,y=0}^{4} (Image[x,y] - (a_0 + a_1 z + a_2 z^2 + a_3 z^3))^2$$

(minimize w.r.t. a_0, a_1, a_2, a_3 and θ)
$z = x.cos(\theta) + y.sin(\theta)$
The initial estimate of θ, from the L.S.E. planar-fit, is refined. The equations to be solved are non-linear in θ.

Least-Squares Criterion for a Tanh-fit :

$$\xi_T = \sum_{x,y=0}^{4} (Image[x,y] - (s.tanh(f[z+p]) + k))^2$$

(minimize w.r.t. s, p and k)
$z = x.cos(\theta) + y.sin(\theta)$
f is determined from the rule of thumb mentioned in section IV i.e. $(0.85 / \sigma_{blur})$. The edge contrast is 2s and p is the position.

Statistics :

ξ_P / σ_{noise} follows the χ^2-Stat. with 22 DOF.
ξ_Q / σ_{noise} approx. follows the χ^2-Stat. with 21 DOF.
ξ_C / σ_{noise} approx. follows the χ^2-Stat. with 20 DOF.
ξ_T / σ_{noise} approx. follows the χ^2-Stat. with 21 DOF.
$\{10.(\xi_P - \xi_C)/\xi_C\}$ approx. follows the 2, 20 F-Stat.
$\{20.(\xi_Q - \xi_C)/\xi_C\}$ approx. follows the 1, 20 F-Stat.

The above formulations are inexact because of the non-linearity of the cubic and tanh bases (in θ and p respectively) and the fact that the value of θ used in the tanh and quadratic fits is predetermined.

Appendix III : Localization of Tanh-Fit

Let $E(x)$ be an ideal step of height S at the origin, $G(x)$ be a normalized gaussian with "standard deviation" σ_{blur} and $\Sigma\,\delta(x-k)$ represent a discrete sampling function.

$$E(x) = \begin{cases} 0 & \text{if } x < 0 \\ S & \text{if } x > 0 \end{cases}$$

$$G(x) = n.e^{-x^2/2\sigma_{blur}^2}$$

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x <> 0 \end{cases}$$

Further, let $\eta(k)$ represent additive white gaussian noise with standard deviation σ_{noise}. Then, we can model a one-dimensional step-edge at position p, as below (where * denotes convolution).

$$f(k) = \left[E(x-p) * G(x)\right]\delta(x-k) + \eta(k) \qquad k = \ldots -1\ 0\ +1 \ldots$$

Let the function fitted to the data, $f(k)$, be $S\{0.5 + 0.5\tanh(\frac{0.85}{\sigma_{blur}}[x-p-\epsilon])\}$, where ϵ is the error in position-localization. The factor 0.85 was chosen to minimize the total-square-error, in the absence of noise. Then, the total-square-error ξ is given by

$$\xi = \sum_k \left[f(k) - S\left\{0.5 + 0.5\tanh\left\{\frac{0.85}{\sigma_{blur}}[x-p-\epsilon]\right\}\delta(x-k)\right\}\right]^2$$

Minimizing ξ w.r.t. ϵ by equating $\frac{\partial \xi}{\partial \epsilon}$ to zero, we get

$$\sum_k \left\{S\,\frac{0.85}{\sigma_{blur}}\,sech^2\left\{\frac{0.85}{\sigma_{blur}}[x-p-\epsilon]\right\}\delta(x-k)\right\}\left[\ \cdots\ \right] = 0$$

where $\left[\ \cdots\ \right]$ *is the error term from the preceding eqn.*

Now, let's assume the signal to have a high S.N.R.. Then, ϵ is small and we can substitute the $sech^2$ and $tanh$ terms by the first two terms of their Taylor series expansion w.r.t. ϵ. Dropping the ϵ^2 term and simplifying, we get

$$\left[\sum_k sech^2\left\{\frac{0.85}{\sigma_{blur}}[x-p]\right\}\delta(x-k)\left\{\Omega(k) + \frac{\eta(k)}{S}\right\}\right]$$

$$+\ \epsilon\left[\sum_k \frac{0.425}{\sigma_{blur}}\,sech^4\left\{\frac{0.85}{\sigma_{blur}}[x-p]\right\}\delta(x-k)\right]$$

$$+\ \epsilon\left[\sum_k \frac{1.7}{\sigma_{blur}}\,sech^2\left\{\frac{0.85}{\sigma_{blur}}[x-p]\right\}\tanh\left\{\frac{0.85}{\sigma_{blur}}[x-p]\right\}\delta(x-k)\left\{\Omega(k) + \frac{\eta(k)}{S}\right\}\right] = 0$$

where

$$\Omega(k) = \left[\frac{1}{S}E(x-p) * G(x) - \left\{0.5 + 0.5\tanh\left\{\frac{0.85}{\sigma_{blur}}[x-p]\right\}\right\}\right]\delta(x-k)$$

Notice that $\Omega(k)$ are samples of the error profile shown in Fig. 5. Hence, the magnitude of $\Omega(k)$ is bounded by 0.01. Invoking the high S.N.R. assumption once again, for typical values of σ_{blur} (≥ 0.5), we can drop the last term by comparison with the other ϵ term. Then,

$$\epsilon = -\frac{\sum_k sech^2\left\{\frac{0.85}{\sigma_{blur}}[x-p]\right\}\delta(x-k)\left\{\Omega(k) + \frac{\eta(k)}{S}\right\}}{\sum_k \frac{0.425}{\sigma_{blur}}\,sech^4\left\{\frac{0.85}{\sigma_{blur}}[x-p]\right\}\delta(x-k)}$$

where $\Omega(k)$ *is as defined above*

Taking the expected value w.r.t. η, the noise, it follows that ϵ, the error, is biased. The bias is a function of the position, p, of the original step and for a typical σ_{blur} ($= 0.6$), it closely resembles a sinusoid with a period of 1 pixel-width and amplitude 1.06E-2 pixel. In practice, we would be required to quantize the position we determine from the tanh-fit and in all likelihood the quantization error will be an order of magnitude more than the bias. Taking the expectation of ϵ^2 w.r.t. η, we get

$$E[\,\epsilon^2 \mid p\,] = \left[\frac{\sum_k sech^2\left\{\frac{0.85}{\sigma_{blur}}[x-p]\right\}\delta(x-k)\,\Omega(k)}{\sum_k \frac{0.425}{\sigma_{blur}}\,sech^4\left\{\frac{0.85}{\sigma_{blur}}[x-p]\right\}\delta(x-k)}\right]^2 + \frac{\left\{\frac{\sigma_{noise}^2}{S^2}\right\}}{\sum_k \left\{\frac{0.425}{\sigma_{blur}}\right\}^2 sech^4\left\{\frac{0.85}{\sigma_{blur}}[x-p]\right\}\delta(x-k)}$$

Now, let's obtain an expression for the root-mean-square of the total-error, taking position-quantization into account. Let the quantization interval, Δ_q, be 0.1 pixel and the quantization levels be centered around the origin. Further, consider p, the actual location of the edge, to be a uniformly distributed random variable in the interval (-0.5 +0.5) and let $\sigma_{blur} = 0.6$. Then, it can be shown that the quantization error is approximately uncorrelated to the bias. Hence, the mean-square of the total-error is the sum of the expectation of $E[\,\epsilon^2 \mid p\,]$ w.r.t. p and $\frac{\Delta_q^2}{12}$. The latter is the variance of the quantization error [15]. For the above choices of σ_{blur} and Δ_q, the root-mean-square error can be numerically evaluated to be $\sqrt{8.9E\text{-}4 + \frac{2.37}{S.N.R.^2}}$. Perhaps, it should be pointed out that the exact choice of the range of summation in the above expressions does not matter, as the first two terms on either side of the origin dominate the calculation. Under simulation with a 5-pixel-width window centered about the origin, this expression was found to be in error by less than 5% for $S.N.R. \geq 8$ and less than 10% for $S.N.R. \geq 4$. Note, that intensity quantization effects were neither accounted for in the analysis, nor present in the simulations.

Acknowledgement

Jim Herson and Cregg Cowan, both at S.R.I. international, made it possible for us to offer a comparison of our results with those of an implementation of the Marr-Hildreth Operator. V.S.N. is indebted to Ron Fearing, who was a constant source of encouragement, and to Brian Wandell for his excellent editorial commments.

References

[1] I.E.Abdou, W.K.Pratt: "Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors," Proc. IEEE, Vol.67, No.5, May 1979, 753-763.

[2] H.C.Andrews, B.R.Hunt: "Digital Image Restoration," Prentice-Hall Inc., Englewood Cliffs, 1977.

[3] T.O.Binford: "Inferring Surfaces from Images," Artificial Intelligence, 17, August 1981, 205-244.

[4] P.Blicher: "Edge Detection and Geometric Methods in Computer Vision," Ph.D. Thesis, Math. Dept., U.C. Berkeley, October 1984.

[5] M.Brady: "Computational Approaches to Image Understanding," Computing Surveys, Vol.14, No.1, March 1982, 3-71.

[6] F.J.Canny: "Finding Edges and Lines in Images" AI-TR 720, M.I.T. A.I. Lab., June 1983.

[7] L.S.Davis: "A Survey of Edge Detection Techniques," Computer Graphics and Image Processing, Vol.4 No.3, Sep. 1975, 248-270.

[8] R.M.Haralick: "Digital Step Edges from Zero Crossing of Second Directional Derivatives," IEEE Trans. PAMI-6, No.1, Jan. 1984, 58-68.

[9] M.H.Hueckel: "An Operator which Locates Edges in Digitized Pictures," Journal of the ACM, Vol.18, No.1, Jan. 1971, 113-125.

[10] Y.Leclerc, S.W.Zucker: "The Local Structure of Image Discontinuities in One Dimension," TR-83-19R, Computer Vision and Robotics Lab., McGill Univ., May 1984.

[11] D.C.Marr, E.Hildreth: "Theory of Edge Detection," Proc. R. Soc. Lond., B 207, 1980, 187-217.

[12] V.S.Nalwa: "On Detecting Edges," presented at the Image Understanding Workshop, New Orleans, Louisiana, Oct. 1984.

[13] R.Nevatia, K.R.Babu: "Linear Feature Extraction and Description," Computer Graphics and Image Processing, Vol. 13, 1980, 257-269.

[14] F.O'Gorman: "Edge Detection using Walsh Functions," Artificial Intelligence, 10, 1978, 215-233.

[15] A.V.Oppenheim, R.W. Schafer: "Digital Signal Processing," Prentice-Hall Inc., Englewood Cliffs, 1975.

[16] J.M.S.Prewitt: "Object Enhancement and Extraction," in Picture Processing and Psychopictorics, B.S.Lipkin and A.Rosenfeld, Eds., Academic Press, N.Y., 1970, 75-149.

[17] K.S.Shanmugam, F.M.Dickey, J.A.Green: "An Optimal Frequency Domain Filter for Edge Detection in Digital Images," IEEE Trans. PAMI-1, No.1, Jan. 1979, 37-49.

[18] K.Turner: "Computer Perception of Curved Objects using a Television Camera," Ph.D. Thesis, A.I. Lab., Univ. of Edinburgh, Nov. 1974.

GC970145-49

Visual Surface Interpolation: A Comparison of Two Methods.

by

Terrance E. Boult†
Columbia University Computer Science Department
NYC, NY 10027. tboult@cs.columbia.edu

§0 Abstract

We critically compare 2 different methods for visual surface interpolation. One method uses the reproducing kernels of Hilbert spaces to construct a spline interpolating the data, such that this spline is of minimal norm. The other method, presented in Grimson (1981), recovers the surface of minimal norm by direct minimization of the norm with a gradient projection algorithm. We present the problem that each algorithm is attempting to solve, then briefly introduce both methods. The main contribution is an analysis of each algorithm in terms of the worst case running time (serial processor), space complexity, and rough estimates of the running time and space costs for massively parallel implementations. We the conclude with a discussion of the differences in the internal representation of the surface in both algorithms.

§1 Introduction.

It has been shown that when presented with sparse depth data (say from random dot stereograms) the human visual system infers a smooth surface passing through these data points. The problem of computer visual surface interpolation is to take a sparse set of depth values and calculate the surface passing through these points that seems to model the surface that humans infer from those same data points. Grimson (1981) presented a computational model of this process in the human visual system, and suggested an algorithm that may be used to recover the perceived surface from the depth data.

Although it may be fruitful from a psychological point of view, to develop algorithms that may be biologically realizable, this restriction may increase the computational cost of the algorithms. Therefore we compare, without regard to biological feasibility, two methods for the solution of this visual surface interpolation problem with the intent to determine which is a more efficient algorithm for use in computer vision.

The first of these methods is that presented in Grimson (1981). Grimson's approach was to represent the surface by a grid of depth values, and to use nonlinear programing techniques and directly minimize the "quadratic variation" or bending energy of the surface. Because the problem was to interpolate the given data, Grimson employed a constrained optimization algorithm called the gradient projection algorithm. Because of this we shall refer to Grimson's approach as the *gradient projection based algorithm*.

†This work supported in part by DARPA grant N00039-84-C-0165 and in part by NSF grant MCS-782-3676

The second method we shall examine is the method of reproducing kernels. This method uses the reproducing kernels of Hilbert or semi-Hilbert spaces to calculate splines of minimal norm. The use of surfaces of minimal norm as the visual surface interpolating the depth data is done in spirit of the minimization approach used in Grimson (1981). The use of reproducing kernels to recover splines of minimal norm is not a new idea. It has been studied by Duchon (1976a, 1976b), Meinguet (1979a, 1979b) and more recently by Franke (1982, 1983), Franke and Neilson (1980) (all but Meinguet called them thin plate splines). However, the method has not previously been given serious consideration for visual surface interpolation – probably because it seems unlikely that the human visual system uses such an approach.

In section 2 we derive an precise formulation of the visual surface interpolation problem. Section 3 presents details of both of the above algorithms. In section 4 we present and compare algorithmic properties (time, space and parallel time complexity, optimality and accuracy of the solution) of both methods. Section 5 is a discussion of the representational advantages of splines in functional form over simply having a grid of depth values. In section 6 we discuss the extensibility of both methods to other spaces of functions, and other norms. Section 7 presents our conclusions.

§2 The Problem.

A naïve formulation of the visual surface interpolation problem might be :

> to find "the best approximation" to a surface using only the knowledge of a number of given points thereon, where we require the surface to be interpolatory, i.e. to pass through all the given data.

A major difficulty with this formulation is that it is not well posed, inasmuch as the information does not uniquely determine a solution. In fact, given any set (of zero measure) of points on a surface there are infinitely many surfaces interpolating those points. To alleviate this problem, we must somehow restrict the class of allowed surfaces and/or give some method of ranking the "plausibility" of a surface.

One of the classical ways of insuring that a problem has a unique solution (applied to visual surface interpolation in Grimson (1981) and Kender, Lee, and Boult (1985)) is to use a functional on the surface as a measure of the "unreasonableness" of the surface, and to restrict the allowed class of surfaces to make it a Hilbert or semi-Hilbert space and make the functional a norm or semi-norm on this space. This formulation insures that there exists a unique solution to the problem of finding a surface from the allowed class which minimizes the functional (and hence is the most reasonable). Throughout this paper we shall *assume* that this type of formulation is appropriate for the problem of visual surface interpolation. We shall not investigate which classes of surfaces are most appropriate, nor which functionals may be good measures of the unreasonableness of a surface. Readers in this aspect of the problem may consult Boult (1986).

In what follows we choose to define "best approximation" in terms on minimal error. We *assume* that error can measured by a norm with respect to the given class of functions. The norm might be the **sup** norm (i.e. the maximal difference between the actual surface and the approximation), or the L^2 norm (integral of the square of the difference at each point). The error may be measured in either a relative (e.g. error of 5%) or an absolute sense (e.g. the surfaces never differ by more than .1 mm) depending on the goals of the user. Finally there error may be measured in the worst case, or on the average (with respect to some measure).

Combining these assumptions a precise formulation of the problem of visual surface interpolation from sparse depth data becomes :

> Let F_1, the space of allowed surfaces, be a Hilbert or semi-Hilbert space. Let F_2 be the elements of F_1 restricted to a finite domain D (since we are only interested in recovering a finite portion of a possibly infinite surface). Let $\Theta(f): F_1 \rightarrow \Re$, be a functional measuring the "unreasonableness" of a surface (i.e. the more reasonable a surface f, the smaller $\Theta(f)$), where Θ is a norm on F_1 (a semi-norm if F_1 is semi-Hilbert). Let $N(f) = \{z_1,\ldots,z_k\} = \{ f(x_1,y_1), \ldots, f(x_k,y_k) \}$ be the allowed information (i.e. the allowed input to solve the problem is k depth values.) Then the *visual surface interpolation problem* is to find (using only N(f)) $f^* \in F_1$, such that
>
> $$\Theta(f^*) = \min_{g \in F_1} \Theta(g).$$

Kender, Lee and Boult (1985) show (as a special case of work on information based complexity see Traub and Woz'niakowski (1980), or Traub, Wasilkowski and Woz'niakowski (1983)) that given the above formulation the surface minimizing the functional $\Theta(f)$ will also be the minimal error surface with respect to the class F_1 for almost *any error* norm.

One functional to measure unreasonableness that is used by both Grimson (1981) (who called it quadratic variation) and Kender, Lee and Boult (1985) is given by :

$$\Theta(f) = \left\| \iint_{\Re^2} \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2 \cdot \left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 \right\|$$

We note that this is just one particular choice for the functional and that this functional is the norm or semi-norm for a number of different classes, see Boult (1985a). It is known that different functionals (and the associated classes for which they are norms) give rise to different interpolation problems, and hence to different interpolating surfaces. The reader interested in other norms and their associated classes should consult Grimson (1981), Boult (1985a) or Boult (1986).

§2.1 Allowed Information.

We now consider the allowed form of the information $N(f) = \{z_1,\ldots, z_k\} = \{f(x_1,y_1), \ldots, f(x_k,y_k)\}$. Each piece of information consists of 2 components, a function value and the location of that evaluation. Throughout this paper we shall assume we are given the value (height above or depth below a reference plane) of the surface at known points in x–y space. Note that this precludes the use of surface gradients, normals, curvature, etc.. This pure depth data might be the result of a stereo based process, a rangefinder or be synthetically generated.

The other component of information is the location of the function evaluations. We consider two separate ways of determining the locations for the information. The first method is to obtain the information from triangulation between matched points in the zero crossings of the Laplacian of the Gaussian of a stereo pair of intensity images (here after $\nabla^2 G$ zero crossing information). This type of information was proposed by Marr and Poggio (1979) as that available in the human visual system, and was used by Grimson in the development of his computational study of surface interpolation in the human visual system. Another method of choosing the location of the information is to use some fixed and regular pattern, e.g. a regular square grid of r points per side, each point separated by a distance h, (thus the number of depth samples is $k = r^2$). This regular grid information would be very difficult, if not totally impossible, to obtain in a passive stereo system but is easily obtained from active

ranging systems. The major difference then between the two types of information is the location of the information samples; which may be effected by the availability of an active ranging system (an option not open to the human visual system).

We note that information derived from the zero crossings of $\nabla^2 G$, yields locations (both the number of and position of) that depend in a very nonlinear way on the surface viewed. This extra information, (i.e. the knowledge that information is evaluated at the location of the zero crossings of the intensity image) is *not* used by any algorithm known to this author. After a casual reading, it might seem that Grimson's algorithm should take advantage of this extra information, inasmuch as the surface consistency constraint Grimson (1981, p.30), shows a relationship between the location of the zero crossings and the variation of a surface. However, Grimson's algorithm is based on the choice a functional (quadratic variation) that does not truly embody the surface consistency constraint, because it minimizes the total variation of the surface and not the variation between zero crossings. Note that it is not necessarily true that the interpolating surface with minimal total variation also has minimal variation between each set of zero crossings. To see this consider a interpolating surface that has almost zero variation between all but one pair of zero crossings (and hence generally satisfies the surface consistency constraint), but whose variation between that pair is arbitrary large (maybe the surface is not even continuous at one of the zero crossings). Such a surface may have arbitrarily large total surface variation but may have minimal variation between zero crossings (except the one pair).

Since neither the reproducing kernel algorithm nor the gradient projection based algorithm make special use of $\nabla^2 G$ type information, we shall freely compare them with respect to both $\nabla^2 G$ zero crossings and regular grid information. Here after, we shall let the number of information points (regardless of its origin) be denoted by k, and the set of information by $N(f) \equiv \{z_1,\ldots z_k\} \equiv \{f(x_1,y_1),\ldots, f(x_k,y_k)\}$.

§2.2 The Desired Output.

The final component in the formalization of the problem is to specify what it means to find an approximation, i.e. we must consider the representation of the desired solution. Though there are many representations we might chose, we shall examine only those two used by the methods under consideration.

The first and simplest representation of the surface is as function values at some predefined points (e.g. on a 2-d mesh). This is the representation used by the gradient projection based algorithm. In this algorithm, the grid is a uniform 2d mesh, large enough to include all information points. We shall let the total number of points in this grid be n.

The other representation of the solution surface we shall consider is as a function of x and y, which can be evaluated at any point. Obviously given this representation the first representation can be recovered but not visa versa.

Note that the user may be interested in recovering the interpolated surface at fewer than the n points used in the first representation. Hereafter let p be the number points at which the interpolatory surface is to recovered. We need not require that the p solution points contain or be contained in the k information points.

Finally we note that it would be improper to compare two different methods if they were calculating the surface in different representations. Therefore, throughout sections 3-6 we shall assume the reproducing kernel method is used first to calculate its spline representation then the spline is evaluated at the p points

the solution is desired at, which is a subset of the n points used in the gradient projection based algorithm.

§3 Description of the Two Methods of Solution.

In this section we briefly describe the two methods of solution to the surface interpolation problem, which we shall be comparing in this paper. We shall refer to the two methods as the *gradient projection based algorithm* and *reproducing kernel algorithm*. We start by discussing the theoretical basis that they have in common.

Neither method actually requires that $\Theta(f)$ be quadratic variation as in (2.1), only that it be a norm or semi-norm on the space F_1. Both methods rely on the theorem from functional analysis that states if $\Theta(f)$ is a norm over the F_1 and F_1 is a Hilbert space then there exists a *unique* function from F_1 minimizing $\Theta(f)$. (If $\Theta(f)$ is a semi-norm and F_1 is only a semi-Hilbert space then the solution exist and is unique up to a member of the null space of $\Theta(f)$.)

The two methods differ in how they minimize the functional $\Theta(f)$ (with respect to the class of functions F_1) and in their representation of the solution.

§3.1 The Gradient Projection Based Algorithm.

We now examine the gradient projection based algorithm as discussed in Grimson (1981). Inherent in the development of this algorithm is the representation as "explicit depth values at all locations within a Cartesian grid of uniform spacing" Grimson (1981, p180). It is also assumed that the information is given at points within this grid, and for simplicity that the the grid is square with size m x m (where $m = \sqrt{n}$). In the following discussion each grid point is represented by its coordinate location (i,j), $1 \le i,j \le m$, and the solution surface is represented as its value at each grid point, i.e. $s_{i,j}$. Grimson begins by deriving a discrete analogue of the functional $\Theta(f)$, and then solves the discrete minimization problem given by: (Equation (3.1))

$$\text{minimize} \quad \sum_{i=1}^{m-2} \sum_{j=0}^{m-1} (s_{i-1,j} - 2s_{i,j} + s_{i+1,j})^2 + \sum_{i=0}^{m-1} \sum_{j=1}^{m-2} (s_{i,j-1} - 2s_{i,j} + s_{i,j+1})^2 + \sum_{i=0}^{m-2} \sum_{j=0}^{m-2} (s_{i,j} - s_{i+1,j} - s_{i,j+1} + s_{i+1,j+1})^2$$

subject to $s_{i,j} = f(i,j) \ \forall f(i,j) \in N(f)$.

To solve this problem he uses a nonlinear programing algorithm called the gradient projection algorithm (actually he seems to use a modified version of this algorithm usually known as Goldfarb's algorithm see Avriel (1976). To implement this algorithm he develops stencils (see Grimson (1981, p183-184)) to allow the calculation of the gradient of the objective function. To determine the amount to move in this direction, one must calculate the minimum of the objective function in that direction. To do this Grimson calculates the value α that minimized the expression: (Equation (3.2))

$$\sum_{i=1}^{m-2} \sum_{j=0}^{m-1} (s_{i-1,j} - 2s_{i,j} + s_{i+1,j} + \alpha d_{i-1,j} - 2\alpha d_{i,j} + \alpha d_{i+1,j})^2 + \sum_{i=0}^{m-1} \sum_{j=1}^{m-2} (s_{i,j-1} - 2s_{i,j} + s_{i,j+1} + \alpha d_{i,j-1} - 2\alpha d_{i,j} + \alpha d_{i,j+1})^2$$

$$+ \sum_{i=0}^{m-2} \sum_{j=0}^{m-2} (s_{i,j} - s_{i+1,j} - s_{i,j+1} + s_{i+1,j+1} + \alpha d_{i,j} - \alpha d_{i+1,j} - \alpha d_{i,i+1} + \alpha d_{i+1,j+1})^2$$

where $d_{i,j}$ is the negative of the value obtained from the convolution of the appropriate stencil (see Grimson (1981, p183-184)) with $s_{i,j}$ (i.e. the negative gradient direction or direction of steepest decent of minimizing the surface variation). He the concludes that $\alpha = \alpha_1 / \alpha_2$ where (Equation (3.3))

$$\alpha_1 = \sum_{i=1}^{m-2} \sum_{j=0}^{m-1} (s_{i-1,j} - 2s_{i,j} + s_{i+1,j})^2 (d_{i-1,j} - 2d_{i,j} + d_{i+1,j})^2$$

$$+ \sum_{i=0}^{m-1} \sum_{j=1}^{m-2} (s_{i,j-1} - 2s_{i,j} + s_{i,j+1})^2 (d_{i,j-1} - 2d_{i,j} + d_{i,j+1})^2$$

$$+ \sum_{i=0}^{m-2} \sum_{j=0}^{m-2} ((s_{i,j} - s_{i+1,j} - s_{i,j+1} + s_{i+1,j+1})^2 (d_{i,j} - d_{i+1,j} - d_{i,j+1} + d_{i+1,j+1})^2)$$

and (Equations (3.4))

$$\alpha_2 = \sum_{i=1}^{m-2} \sum_{j=0}^{m-1} (d_{i-1,j} - 2 d_{i,j} + d_{i+1,j})^2$$

$$+ \sum_{i=0}^{m-1} \sum_{j=1}^{m-2} (d_{i,j-1} - 2 d_{i,j} + d_{i,j+1})^2$$

$$+ \sum_{i=0}^{m-2} \sum_{j=0}^{m-2} (d_{i,j} - d_{i+1,j} - d_{i,j+1} + d_{i+1,j+1})^2$$

Thus the complete gradient projection based algorithm employed by Grimson consists of the following 5 steps:

Step 1: Determine a feasible initial surface (any surface interpolating the information will do).

Step 2: Compute the negative of the gradient direction (the $d_{i,j}$ above) by taking the the convolution of the current approximation (the $s_{i,j}$'s) with the stencils (setting the $d_{i,j} = 0$ if i,j is an information point).

Step 3: Compute α_1 and α_1 (from formulas (3.3) and (3.4) above) and then set $\alpha = \alpha_1 / \alpha_2$.

Step 4: Refine surface approximation (i.e. for each i,j set $s_{i,j} := s_{i,j} + \alpha \cdot d_{i,j}$).

Step 5: If $d_{i,j} \leq \varepsilon \ \forall \ i,j \leq m$ then approximation is complete
Else goto Step 2;

§3.2 The Method of Reproducing Kernels.

The method of reproducing kernels calculates a spline function that *exactly* solves the continuous problem of finding the function from F_1 minimizing $\Theta(f)$. There are at least two different algorithms based on the use of reproducing kernels, we shall present only one. The interested reader may consult Boult (1985b) or Boult (1986) for a more detailed discussion of both algorithms based on reproducing kernels. The following discussion of reproducing kernels for interpolation is based on the theoretical work of Meinguet (1979a, 1979b).

For this method to be appropriate it is sufficient to have F_1 be a semi-Hilbert space and $\Theta(f)$ the associated semi-norm with null space Π_1. (Throughout this paper Π_1 is the space spanned by $\{1,x,y\}$). To insure uniqueness of the solution we must assume that the information $N_k(f)$ contains a $\Pi_1$unisolvent subset, i.e. there exists a set J of indices (a subset of the index set I = 1... k)

and associated information points x_j,y_j with information values z_j such that for each element of Γ (there are 3 in the present case) there exists a unique $p_j(x,y) \in \Pi_1$ such that for all j, j' from J, $p_j(x_j,y_j) = 1$ and $p_j(x_{j'},y_{j'}) = 0$ if $j \neq j'$. Note that if N_k contains evaluations 3 or more non-colinear points, then $N_k(f)$ will contain a Π_1 unisolvent subset. (This restriction on the information having at least 3 non-colinear points also applies to the gradient projection based algorithm.)

In the development of this method we use that fact that we can separate the space F_1 into $X_0 \oplus \Pi_1$, $X_0 = \{g \in F_1 : g(x_j,y_j) = 0, \forall\, j \in J\}$ where $\oplus$ is a (topological) direct sum. With this decomposition X_0 is a Hilbert space with $\Theta(\cdot)$ as a norm (not a semi-norm). Given the reproducing kernel $K_0((s,t);(x,y))$ of X_0 (which can be expressed in terms of the reproducing kernel of F_1 and the functions $p_j(x,y)$ see Boult (1985b)) the spline surface, of minimal $\Theta(f)$ norm, which interpolates the information $N(f)$ is given by (Equation (3.5) :

$$s_N(x,y) = \sum_{i \in I-J} \gamma_i \; K_0((x_i,y_i);(x,y)) + \sum_{j \in J} z_j \cdot p_j(x_j,y_j)$$

where the coefficients γ_i can be calculated from the (k-3) by (k-3) dense linear system (equation 3.6):

$$\sum_{i \in I-J} K_0(x_k,y_k;x_i,y_i) \cdot \gamma_i = z_i - \sum_{j \in J} z_j \; p_j(x_j,y_j) \quad \forall\, k \in I.$$

The reproducing kernel method then consists of the three following steps:

Step 1: Calculate the matrix of coefficients for the left hand side of equation (3.6).

Step 2: Compute γ_i, i =1..k-3, the solution to equation (3.6).

Step 3: Compute the value of interpolating surface at all solution points using equation (3.5).

Note that Step 1 and 2 are necessary parts of the algorithm, whereas step 3 is simply to allow comparison of this method with the gradient projection based algorithm. Also note that for fixed regular data it is possible to precompute the Cholesky decomposition of the coefficient matrix (which is determined entirely by the location of the information), and then step 2 is simply the calculation of the γ_i's using back substitution.

A proof that the above spline is of minimal norm can be found in Meinguet (1979a, 1979b).

§4 Comparison of Computational Issues of The Two Methods.

In this section we provide an analysis and comparison the two visual surface interpolation methods on a number of computational issues. These issues and the subsection in which they are treated are:

§4.1 Time complexity,

§4.2 Space complexity,

§4.3 Inherent Parallelism and Parallel Time Complexity,

§4.4 Optimality and Accuracy of Solution.

For all of the comparisons we shall assume that k, p, n are defined as in sections 2 and 3. When we refer to the steps of the gradient projection based algorithm and the reproducing kernel method, we are referring to the steps as defined in sections 3.1 and 3.2 respectively. A synopsis of the results can be found in Table 4.1.

§4.1 Time Complexity.

First lets us estimate an upper bound on the worst case running time (assuming each arithmetic operation costs unity) of the reproducing kernel method when the information is from $\nabla^2 G$ zero crossings. Obviously step 1 costs $O(k^2)$, and step 3 costs $O(kp)$. For step 2, using Cholesky decomposition (the matrix is positive definite), the cost will be $.5k^2$. Therefore the worst case cost is $.5k^2 + O(k^2 + kp)$. (A careful analysis of the current implementation results in a cost of $\equiv .5k^2 + 70kp$: $O(k^2)$, but this depends on the choice of the space F_1, norm $\Theta(f)$ and the associated reproducing kernel.)

Now we note that if the information is gathered on a regular grid, then we can do the Cholesky decomposition once (a precomputation), and store the results. Given this decomposition we can reduce the cost of step 2 to $O(k^2)$, and the overall cost to $O(k^2 + kp)$. (In fact, given the decomposition for a grid of size r x r we also have the decomposition for all smaller grids.)

Now let us estimate an upper bound on the worst case running time of the gradient projection based algorithm. Step 1 of that algorithm obviously cost O(n). Examination of the stencils given in Grimson (1981, p183) yields a cost per iteration for step 2 of approximately 26n. For step 3, equations (3.3) and (3.4) yield a per iteration cost of approximately 24n. Finally for step 4 costs 2n per iteration. Thus the total per iteration cost of the algorithm is $\equiv 50n$. We take the number of iterations to be n, which is the upper bound on the number iteration of the Gold - farb's algorithm (the nonlinear programming algorithm on which the method is based) for problems of this type (see Avriel (1976, p436)). (Note that this is far better than the $O(n^2)$ iterations which Terzopoulos (1984, p103) suggests the gradient projection based algorithm takes.) Combining the above we arrive at a total estimated cost for the gradient projection based algorithm of approximately $50n^2$. (We note that the number of iterations will actual depend on the number of, value of, and location of the information. Thus one may be able to get a better estimate for fixed regular data. However one can easily show that no placement of data can result in less than $\sqrt{(n/k)}$ iterations and worst case placement of fixed data can easily be shown to result in at least $2\sqrt{n}$ iterations. Both of these bounds are trivial, and the actual lower bound is probably O(n).)

Thus for $\nabla^2 G$ zero crossing information the reproducing kernel method is faster whenever $.5k^2 + 70kp < 50n^2$. And for grid data, the reproducing method is faster when $O(k^2) + 70kp < 50n^2$.

§4.2 Space Complexity.

The space required for the reproducing kernel method is $.5k^2 + O(k)$ for steps 1 and 2 (independent of the type of information), and $k+p + O(1)$ for step 3. (This assumes that the user needs all p values at the same time. If the user can use the points sequentially, then the space for step 3 is simply $k + O(1)$.)

The space complexity of the gradient projection based algorithm can be calculated by examining equation (3.2). Although algorithms trying to obtain minimal time complexity may use more space, each iteration of the algorithm can be programed using only $2n + O(\sqrt{n})$ space. Note that no savings in space is obtained if the user only requires the solution points one at a time.

Therefore the reproducing kernel method will use less space whenever $\min(k^2, k+p) < 2n$, (i.e. whenever $k < \sqrt{(2n)}$ because $p \leq n$).

§4.3 Inherent Parallelism and Parallel Time Complexity.

In this subsection we examine the sources of parallelism in both of the methods, and estimate their parallel time complexity. True values may vary depending on the instruction set of the parallel machine being used, its topology, its memory limitations, number of processors and its mode of operation (SIMD or MIMD).

There are four different sources of parallelism in the reproducing kernel method. The first is the evaluation of the spline function at one point, which involves the evaluation and summation a weighted kernel function at each of the k information points. This can be parallelized in a straight forward SIMD fashion to run in time O(log k).

The second source of parallelism in the reproducing kernel method is the evaluation of the p surface solution points. These points can easily be evaluated simultaneously, again in a SIMD fashion, resulting in a factor of p speedup. Combining the first two parallelizations we could speed up the surface reconstruction (given the coefficients of the spline) from 70kp to O(log k).

The third form of parallelism come from the calculation of the coefficients of the spline. Given the decomposition of the coefficient matrix, we can compute the coefficients in parallel in time k.

The final type of parallelism is that inherent in the solution of a kxk linear system. This has been studied elsewhere and in general one can gain a speed up factor of k.

Thus our estimate of the parallel running time for the reproducing kernel algorithm is $O(k^2)$ if we must decompose the matrix coefficients (as we must for $\nabla^2 G$ information), and O(k) if the decomposition is precomputed (as it is may be for regular gird data).

The parallelisms inherent in the gradient projection based algorithm include the calculation of the gradient direction, local calculation of each of the terms needed for the calculation of the parameter α, and updating the surface. These will reduce the number of operations per iteration by an estimated amount of 26, 20, 2 respectively. Furthermore, given the local terms for the calculation of the parameter α, we can speed up that calculation by a factor of (log n) / n by using log reduction for the summation. Note that the number of iterations cannot be reduced by parallel implementation. The total estimated time complexity of the parallel gradient projection based is $O(n \cdot \log n)$.

Based on these estimates, a parallel implementation of the reproducing kernel method would be faster than a parallel implementation of the gradient projection based algorithm when $O(k^2) < O(n \cdot \log n)$ and the location of the data is allowed to vary, or when the data is fixed and $O(k) < O(n \cdot \log n)$.

§4.4 Optimality and Accuracy of Solution.

Both methods under consideration started off with the idea of finding the surface of minimal norm ($\Theta(f)$) over F_1, a Hilbert space (or semi-Hilbert space). This had the advantage of resulting in a unique specification of the surface to recover. As mentioned before it is known that such a surface is also a minimal error solution among all surfaces in the class F_1 that interpolate the data, and that this minimal error property holds for almost any reasonable definition of error, see Kender, Lee, and Boult (1985). Thus theoretically both methods are attempting to find an optimal error interpolant from F_1.

The reproducing kernel method theoretically does calculate this optimal error surface. The errors in the coefficients of the spline surface, introduced by the approximate solution of the linear system, however result in the algorithm reconstructing a different surface. The magnitude of the error in these coefficients depend on the condition number of the linear system, which in turn depends on the placement of the information points. Initial experiments suggest that for a regular grid of information the condition number is approximately 19.5 k^2. Because Cholesky decomposition and back substitution are numerically stable (given proper implementation), we know the resulting coefficients differ from the true spline coefficients by at most $e_1 \equiv c_1 \cdot 2^{-t} \cdot 19.5\ k^2$, where t is the number of bits in the mantissa of the floating point representation on the machine and c_1 is a fixed constant depending of the floating point implementation. Then the maximum error of any surface reconstruction (from the optimal surface, not from the surface generating the information) is $< k \cdot e_1 \max(K_0)$. Note that this is totally independent on the number of reconstruction points, but depends on the distance of the reconstructed points from the information points. ⊙

The gradient projection based algorithm however leaves its theory behind. The first step in the method is the discretizaton of the functional to minimize. This discretizaton is well studied in mathematical physics, and the error introduced by it is $O(h^2)$ where h is the distance between grid points. The method then attempts to minimize this discrete functional without regard for the space F_1, thus its solution may not even be a "feasible solution". (Note that this is not as simple a problem to over come as it might seem because the discretized version of $\Theta(f)$ is no longer a norm or semi-norm on F_1 so there is not even an assurance of a surface minimizing the discretized version of $\Theta(f)$ existing, let alone being reachable by a sequence of surfaces from F_1.) Finally there is the error introduced by the gradient projection portion of the algorithm, and by terminating the algorithm before it has computed the exact solution (to the perturbed problem.) Currently we do not have estimates on the error of the algorithm, but the work of Terzopoulos (1984) suggests that the error does goes to zero, albeit very slowly, as both the number of points and number of iterations grows.

	Reproducing(F)	Reproducing(V)	Gradient Projection
TC	$O(k^2 + kp)$	$(k^2/\epsilon + O(k^2 + kp)$	$50\ n^2$
SC	$(.5k^2 + O(kp))$	$(.5k^2 + O(kp))$	$(2n + O(\sqrt{n}))$
PT	$O(k^2)$	$O(k)$	$O(n \log n)$

Table 4.1. Comparison of essential properties of 2 algorithms. TC stands for time complexity, SC for space complexity, and PT for parallel time complexity. Here k is the number of information (depth) samples; p is the number of points in the desired solution; n is the number of points in the grid used by the gradient projection based algorithm. The F and V in the titles refer to fixed and varying data respectively where fixed and varying data refer *only* to the location of the information points, which effect the performance of the reproducing kernel algorithm.

§5 Advantages of Spline Representation.

In this section we discuss the advantages of the spline representations over the mesh / grid representation used by the gradient projection based algorithm.

The first advantage of the functional spline representation (as a weighted sum of kernel functions) is that we can easily compute a estimate of any functional on the "true surface" (e.g. an integral or a derivative of the surface that generated the information

points) by applying said functional to the spline as a function. In fact, provided that the functional is linear and that the space F_1 is sufficiently smooth, the estimation so obtained is an optimal error estimate (see Traub and Woźniakowski (1980)). Using this fact, we can easily compute the orientation of the surface at any point, and even estimate the bending energy (however since this is not a linear functional, it it not necessarily an optimal estimate). These values might be used to segment the image, or locate surface discontinuties.

A second advantage of the spline representation is that it can easily be used in a system that has a focus of attention. It can easily generate depth values at any points, and if the system decides (after looking at some initial depth values) that it would like to look at a portion of the surface in more detail, there is no need to recalculate the spline, simply evaluate the spline function at the new desired points. Along these lines, the system can also update its idea of a surface, by adding a new information point, calculating the updated spline coefficients (this costs only k^2) and updating the surface points.

A third advantage of this representation is that it is less orientation dependent than a grid of values. If the visual system were to rotate or translate (as long as it does not change the relative order and spacing of the information points) then the coefficients of the spline are the same, and the spline is simply rotated or translated as well. This property does not hold for a grid of data.

A final advantage is that this representation is generally more compact (in space terms) than the grid representation. The spline is defined by 3k values, these are the k coefficients and the location of the k information points. Given these values, one can reconstruct the spline or compare this spline with another spline. This could then be used as a means of saving the reconstructed surface. Also given advantage three above, the spline representation might be used in a surface recognition algorithm.

§6 Extensibility of the Interpolation Methods.

In Grimson (1981), Grimson argued, rather convincingly, that the correct "unreasonableness" functional was quadratic variation. He however considered only a particular form of functional, and there may be more appropriate functionals which are not of the form he considered. Also, the space in which we attempt to minimize the functional has some effect on the interpolating surface, and we should consider other possible spaces for reconstruction, even for the quadratic variation functional.

Inasmuch as the reproducing kernel method finds the surface of minimal norm from a Hilbert space (or semi-norm and semi-Hilbert), in theory it can be applied to any "unreasonableness" functional which is the norm (semi-norm) of such a space. However, to do this we must be able to construct the reproducing kernel of said space, and this can be technically *very* difficult. Thus we can easily apply it only in those situations when the reproducing kernel is already known. Fortunately there are a number of such spaces some of which may be appropriate for visual surface reconstruction, see Boult (1985a). In fact, a number of these kernels exist for higher dimensions allowing the algorithm to be extended into arbitrary dimensions.

Given the different reproducing kernel, the only change to the algorithm is to replace all evaluations of the old kernels with the appropriate evaluations of the new kernel. If the null space of this new space is different from that of the space F_1, then we must also change the functions $p(x,y)$ used by the algorithm.

To extend the gradient projection based algorithm to another functional, one would first have to develop the new discretized version of the functional. Given this discretization, one would then derive how to compute the negative of the gradient function, and the parameter α. Note that if the functional is not quadratic, these last two modification may be very difficult. Modifying the algorithm to compute the minimization with respect to another class of functions is not even possible since it approximates the surface without regard to a space of functions. It would also be very difficult if not impossible to add this feature into the algorithm, since it would involve verifying that the surface produced by each iteration was a member of the given class of functions.

§7 Conclusions.

In this paper we have compared two different algorithms for visual surface interpolation. And with the possible exception of biological feasibility, we found that if the information was sparse, the reproducing kernel algorithm surpassed the gradient projection based algorithm in almost every important algorithmic aspect. If, however, the number of information points was comparable to the number of points at which we are to recover the interpolation surface, then the gradient projection based algorithm may be superior.

Given that the problem of visual interpolation is generally posed as one with sparse information, we believe that the reproducing kernel method will be a superior algorithm for computer vision uses.

§8 Acknowledgements.

I would like to thank David Lee and John Kender, both of whom played major roles in the current investigaton, and into the application of the reproducing kernel method to visual surface interpolation.

§9 References.

Boult, Terrance, (1985a). Smoothness Assumptions in Human and Machine Vision: Their Implications for Optimal Surface Interpolation, Columbia University, Computer Science Department Technical Report.

Boult, Terrance, (1985b): Reproducing Kernels for Visual Surface Interpolation, Columbia University Computer Science Department Technical Report.

Boult, Terrance, (1986): Information Based Complexity. Applications in Nonlinear equations and Computer Vision, Doctoral dissertation, in preparation.

Franke, R., and Nielson, G. (1980): Smooth Interpolation of Large Sets of Scattered Data, *International Journal for Numerical Methods in Engineering*, Vol 15, 1691-1704.

Franke, R. (1982): Scattered Interpolation: Tests of Some Methods, *Mathematics of Computation*, 38 #157, 181-200.

Franke, R. (1984): Thin Plate Splines with Tension, to appear *CAGD*.

Grimson, W.E.L., (1981): *From Images to Surfaces: A Computational Study of the Human Early Visual System*, MIT Press, Cambridge, MA.

Kender, John, David Lee and Terrance Boult, (1985): Information Based Complexity Applied to the 2 1/2 D Sketch', *Proceedings of the Third IEEE Workshop on Computer Vision: Representation and Control*, p157-167.

Marr, David and Thomas Poggio, (1979): A computational Theory of Human Stereo Vision, *Proc. R. Soc. Lond.* B 204, 301-328.

Meinguet, Jean, (1979a): Multivariate Interpolation at Arbitrary Points Made Simple, *Journal of Applied Mathematics and Physics 30*, 292-304.

Meinguet, Jean, (1979b): Basic Mathematical Aspects of Surface Spline Interpolation, *ISNM 45: Numerische Integration*, 211-220, G. Hämmerilin ed., Basel: Birkhäuser Verlag.

Terzopoulos, Demetri, (1983): Multi-level Computational Processes for Visual Surface Reconstruction, *Computer Vision, Graphics, and Image Processing* 24, 52-96.

Traub, J.T. and H. Woźniakowski, (1980): *A General Theory of Optimal Algorithms*, Academic Press NY.

Traub, J.T., G. Wasilkowski, and H. Woźniakowski, (1983): *Information, Uncertainty and Complexity*, Addison Wesley, MA.

CiC97C145-58

PREDICTING SPECULAR FEATURES

Glenn Healey and Thomas O. Binford

Artificial Intelligence Laboratory
Stanford University
Stanford, California 94305

Abstract

We show that highlights in images of objects with specularly reflecting surfaces provide significant information about the surfaces which generate them. A brief survey is given of specular reflectance models which have been used in computer vision and graphics. For our work, we adopt the Torrance-Sparrow specular model which, unlike most previous models, considers the underlying physics of specular reflection from rough surfaces. From this model we derive powerful relationships between the properties of a specular feature in an image and local properties of the corresponding surface. Careful experiments with specularly reflecting objects establish the merit of these relationships.

1. Introduction

Shiny surfaces give us specular reflections (highlights). A perfectly smooth shiny surface (e.g. a perfect mirror) reflects light only in the direction such that the angle of incidence equals the angle of reflection. For rougher shiny surfaces (e.g the surface of a metal fork), specular effects are still observable. In this paper we analyze the properties of specular reflection from rough shiny surfaces.

There are several basic reasons why the study of specular reflection deserves serious attention in computer vision. Specular features are almost always the brightest regions in an image. Contrast is often very large across specularities; they are very prominent. This makes them easy to locate. In addition, the presence or absence of specular features provides immediate constraints on the positions of the viewer and light sources relative to the specular surface. Also, as we will show, the properties of a specularity constrain the local shape and orientation of the specular surface.

An ability to understand specular features is valuable for any vision system which is required to analyze images of shiny objects. This work, for example, began as an attempt to allow ACRONYM [3] to reason about specular reflections from shiny mechanical parts in the ITA project [4]. Images of these parts typically contain large specular regions. The recognition task becomes considerably easier if the system is able to predict the characteristics of these specular regions.

In this paper we examine what information can be inferred from an image of a rough shiny surface by considering only the physics of specular reflection. Particular emphasis is placed on finding symbolic quasi-invariant relationships which will hold in many different situations (e.g. different source, viewer configurations). In contrast to many intensity-based vision algorithms, our relationships are based on the properties of a relatively large number of pixels in an image. This allows us to observe predicted features and infer local surface shape even in noisy intensity images or in cases where available specular models do not completely characterize the physics of specular reflection.

2. Review of Previous Work

Researchers in computer graphics have used increasingly realistic specular models. Several of these models will be discussed in the next section. In computer vision, however, relatively few attempts have been made to exploit the information encoded in specularities. Ikeuchi [9] employs the photometric stereo method [15] and uses a distributed light source to determine the orientation of patches on a surface. Grimson [8] uses Phong's specular model [10] to examine specularities from two views in order to improve the performance of surface interpolation. Coleman and Jain [5] use four-source photometric stereo to identify and correct for specular reflection components. In more recent work, Blake [1] assumes smooth surfaces and single point specularities to derive equations to infer surface shape using specular stereo. He shows that the same equations can be used to predict the appearance of a specularity on a smooth surface when using a distributed light source.

Takai, Kimura, and Sata [13] describe a model-based vision system which recognizes objects by predicting specular regions. As specular models and insights improve, we expect to see more work which makes use of the properties of specular reflection.

3. Specular Reflectance Models

Given a viewer, a surface patch, and a light source, a reflectance model quantifies the intensity the viewer will perceive. The most general reflectance models represent the perceived intensity I as a sum of three independent reflection components

$$I = I_A + I_D + I_S \qquad (1).$$

Here I_A represents the ambient reflection, I_D represents diffuse (Lambertian) reflection, and I_S represents specular reflection. In this paper, we restrict our attention to the I_S reflection component.

We note that it is typically very easy to separate the I_S reflection component from the I_A and I_D reflection components in an image. There are two distinctive properties of specular reflection. First, over most of a surface I_S is zero, but in specular regions I_S is usually very large relative to I_A and I_D. Secondly, in regions where the specular component is nonzero, I_S changes much more rapidly than either I_A or I_D.

Before discussing the various specular reflectance models, we introduce the reflection geometry (Figure 1). We consider a viewer looking at a surface point P which is illuminated by a point light source. Define

$\vec{V}$	unit vector from P in direction of viewer
N	unit surface normal at P
L	unit vector from P in direction of source
H	$\frac{V+L}{\lvert V+L\rvert}$ (unit angular bisector of $\vec{V}$ and $\vec{L}$)
α	$\cos^{-1}(\vec{N}\cdot\vec{H})$ (the angle between $\vec{N}$ and $\vec{H}$)
θ	$\cos^{-1}(N\cdot V)$ (the angle between $\vec{N}$ and $\vec{V}$)

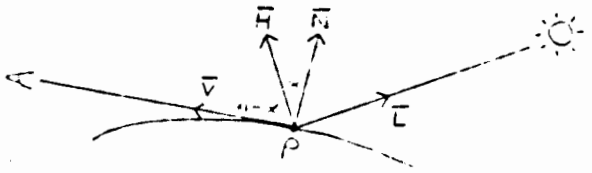


Figure 1. The Reflection Geometry

Throughout this paper, we consider only illumination from a single point light source. In principle, we lose no generality using this kind of an approach since we can describe distributed light sources as arrays of point sources. Thus to handle situations involving distributed light sources we only need to integrate the effects of an equivalent array of point sources.

The simplest specular model assumes that specularities only occur where the angle between $\vec{L}$ and $\vec{N}$ equals the angle between $\vec{N}$ and $\vec{V}$ and $\vec{L}$, $\vec{N}$, and $\vec{V}$ all lie in the same plane. This corresponds to the situation $\alpha = 0$ in Figure 1. Unless the surface is locally flat, this model predicts that specularities will only be observed at isolated points on a surface. A few experiments, however, show that this model is inadequate for most real surfaces. Not only are observed specular features usually larger than single points, but highlights often occur in places which are not predicted by this model.

An empirical model for specular reflection has been developed by Phong [10] for computer graphics. This model represents the specular component of reflection by powers of the cosine of the angle between the perfect specular direction and the line of sight. Thus Phong's model is capable of predicting specularities which extend beyond a single point. While Phong's model gives a reasonable first approximation which is useful in many practical situations, it is possible to develop more accurate models by examining the physics underlying specular reflection.

The Torrance-Sparrow model [14], developed by physicists, is a more refined model of specular reflection. This model assumes that a surface is composed of small, randomly oriented, mirror-like facets. Only facets with a normal in the direction of $\vec{H}$ contribute to I_S. The model also predicts the shadowing and masking of facets by adjacent facets using a geometrical attenuation factor. The resulting specular model is

$$I_S \quad \frac{FDG}{\vec{N}\cdot\vec{V}} \qquad (2)$$

where

F	Fresnel coefficient
D	facet orientation distribution function
G	geometrical attenuation factor

We will analyze the effects of each factor in the model in the next few paragraphs. The results we present in this paper are derived from (2).

The Fresnel coefficient F models the amount of light which is reflected from individual facets. In general, F depends on the incidence angle and physical properties

of the reflecting surface. Cook and Torrance [6] have shown that to obtain realistic graphics images, F must characterize the color of the specularity. For metal surfaces F is approximately a constant.

The distribution function D describes the orientation of the micro facets relative to the average surface normal $\bar{N}$. Blinn [2] and Cook and Torrance [6] discuss various distribution functions. In agreement with Torrance and Sparrow we use the Gaussian distribution function given by

$$D = Ke^{-(\alpha/m)^2} \quad (3)$$

where K is a normalization constant. Thus for a given α, D is proportional to the fraction of facets oriented in the direction $\bar{H}$. The constant m indicates surface roughness and is proportional to the standard deviation of the Gaussian. Small values of m describe smooth surfaces for which most of the specular reflection is concentrated in a single direction. Large values of m are used to describe rougher surfaces with large differences in orientation between nearby facets. These rough surfaces produce specularities which are spread out on the reflecting surface.

The expression for the geometrical attenuation factor G is derived by Torrance and Sparrow in [14]. They assume that each specular facet makes up one side of a symmetric v-groove cavity. From this assumption, they examine the various possible facet configurations which correspond to shadowing or masking. They quantify the geometrical attenuation factor as

$$G = \min\left\{1, \frac{2(\bar{N}\cdot\bar{H})(\bar{N}\cdot\bar{V})}{(\bar{V}\cdot\bar{H})}, \frac{2(\bar{N}\cdot\bar{H})(\bar{N}\cdot\bar{L})}{(\bar{V}\cdot\bar{H})}\right\} \quad (4).$$

We will show that in applications it is often possible to use a simpler expression for G.

As θ increases from 0 to $\frac{\pi}{2}$, the viewer gradually sees a larger part of the reflecting surface in a unit area in the view plane. Therefore, as θ gets larger, there are correspondingly more surface facets which contribute to the intensity perceived by the viewer. We take this phenomenon into account in (2) by dividing by $\bar{N}\cdot\bar{V}$

4. Inferring Surface Properties

In this section, we demonstrate how we can use (2) to determine local surface properties from specularities. In almost all situations we do not require the full generality of (2) to infer these local properties. Our first assumption is that F is constant. This is a very good approximation for most specularly reflecting surfaces. We can further simplify (2) by observing that except for a small range of angles near grazing incidence, the value of G is unity. We will discuss this result and the exceptional cases later. Hence the form of (2) used to determine local surface properties is

$$I_S = \frac{Ke^{-(\alpha/m)^2}}{(\bar{N}\cdot\bar{V})} \quad (5).$$

Referring again to the geometry of Figure 1, we assume that the viewer and light source are distant relative to the surface. Therefore $\bar{V}$ and $\bar{L}$ are essentially constant and hence their angular bisector $\bar{H}$ is essentially constant. We assume that the positions of the viewer and light source are know . Finally, since the distance from the viewer to the surface is large, we can approximate the perspective projection of the imaging device with an orthographic projection.

Doubly Curved Surfaces

For a surface which is doubly curved at a specularity (i.e. both principal curvatures are different from zero) we will be able to locate a single point P_0 of maximum intensity in the image of the specularity. From (5) we see that this point corresponds to the local surface orientation $\bar{N} = \bar{H}$ (i.e. $\alpha = 0$). Given a doubly curved surface where $\bar{H}$ is known, we can very quickly determine the surface orientation at P_0.

Figure 2 shows a typical specular image generated about an elliptic point on a surface.

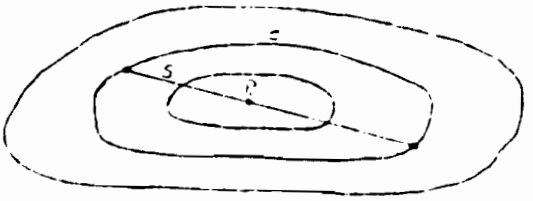


Figure 2

Specular Intensities on a doubly curved surface

The closed curves are image curves of constant intensity. P_0 corresponds to $\alpha = 0$. As predicted by (5), intensity decreases as we move away from P_0. Now suppose we examine a straight segment S in the image which intersects P_0 and which begins and terminates on a constant

intensity curve C. Let the measured intensity at P_0 be I_0 and let the measured intensity on C be I_1. Since the exponential factor in (5) will change very fast relative to the change in $(\bar{N} \cdot \bar{V})$, we can consider $(\bar{N} \cdot \bar{V})$ to be constant on S. Define $d\alpha$ to be the change in α as we move from the surface point imaging to P_0 to the surface point imaging onto C along S.

If we let $K' = \frac{K}{(\bar{N} \cdot \bar{V})}$, then from (5) we have

$$I_0 = K', \qquad I_1 = K' e^{-(\alpha/m)^2} \quad (6)$$

which give us

$$d\alpha = m\sqrt{\ln I_0 - \ln I_1} \quad (7)$$

To determine the surface curvature along S we need to compute the arc length of the curve on the surface which generated S in the image. This will depend on $\bar{V}$. We do this by introducing an x,y,z coordinate system such that the surface curve S' imaging to S lies in the x-y plane (Figure 3). Denote by P_0' the point imaging to P_0 and place P_0' at the origin. Further arrange the coordinate system so that $\bar{H}$ is parallel to the y axis. Let $\bar{V}'$ be the projection of $\bar{V}$ onto the x-y plane.

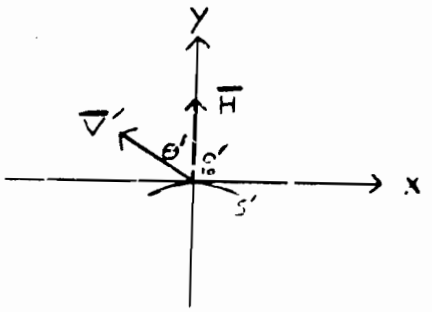


Figure 3. The Surface Geometry

Since the specularity will only be observable for a small range of α, we can approximate the arc length of S' by the length of its projection onto the tangent plane to the surface at P_0'. If l_S denotes the arc length of S' and l_I denotes the length of S in the image then letting θ' be the angle between V' and the y axis we have

$$l_S = \frac{l_I}{\cos\theta'} \quad (8).$$

Using (7) and (8) we can estimate the curvature κ on S' at P_0' as

$$\kappa = \frac{2d\alpha}{l_S} \quad (9)$$

Note that here we are using the fact that since the length of S' is small the surface normal at any point on S' lies approximately in the same plane as S' and $\bar{H}$ in Figure 3.

For any segment like S which intersects P_0 and begins and terminates on C we can compute a corresponding value of κ. If we examine these line segments for every direction in the image, then on the tangent plane to the surface at P_0' we will be examining the corresponding line segments $l_1,...,l_n$ in every direction through P_0'. Consequently, on the surface we will compute κ for all curves formed by intersecting the surface with planes containing $\bar{N}$ and l_i (for i=1,...,n). The largest and smallest computed values of κ will give the principal curvatures of the surface at P_0' [12]. Therefore we can determine the principle curvatures and the principle directions at P_0'.

Singly Curved Surfaces

If one principal curvature of a surface is zero in a specular region we will not be able to immediately infer the local orientation as we did for a doubly curved surface. To understand why, consider Figure 4. Figure 4 shows a viewer looking at a tilted cylinder. To make the example concrete, assume that $\bar{L}$ is such that $\bar{H} = \bar{V}$.

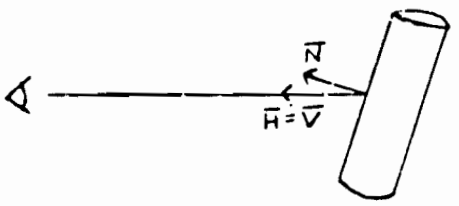


Figure 4. Viewer Observing a Tilted Cylinder

For this configuration there will be no point on the surface for which $\alpha = 0$ (recall that $\bar{H}$ is essentially constant), yet we will still observe a specularity in the image if at some point α is small enough to give a significant value for I_S in (5). The lines of constant I_S in the image appear as in Figure 5. Here C_0 corresponds to the

smallest α and therefore the largest specular intensity. As predicted by the model, I_S decreases as we move away from C_0.

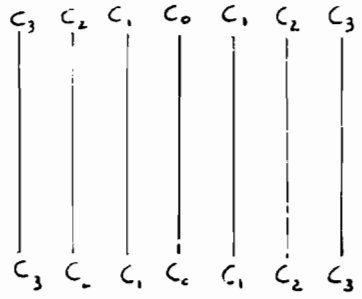


Figure 5. Lines of Constant I_S for a Cylinder

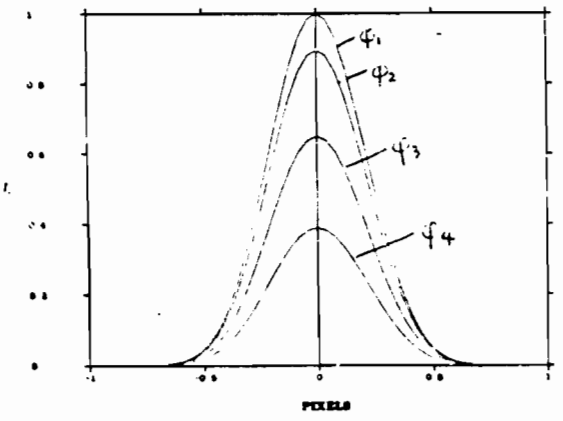


Figure 6. I_S for different values of ϕ

We observe that it is typically easy to detect the fact that a surface is singly curved at a specularity. This is because we will observe a line of maximum intensity (along the line of zero curvature) instead of the point maximum we observe for the doubly curved case.

Next we examine how we can infer curvature and orientation from an image of a singly curved surface. We cannot simply apply the analysis for the doubly curved case since in general we do not know α at surface points on the line of maximum specular intensity. In Figure 3 it is reasonable to assume that $z = 0$ at P_0'. For the singly curved case, however, the relationship will be more complicated. Modify the geometry of Figure 3 so that $\bar{H}$ lies in the y-z plane and makes an angle ϕ with the x-y plane. The normal to S' at P_0' is still considered to be parallel to the y axis. If the curvature of S' at P_0' is $1/r$ then locally we have

$$\bar{N} = \left(\frac{x}{r}, \frac{\sqrt{r^2 - x^2}}{r}, 0\right) \quad (10)$$

$$\bar{H} = (0, \cos\phi, \sin\phi) \quad (11)$$

$$\alpha(x) = \cos^{-1}\left(\frac{\cos\phi}{r}\sqrt{r^2 - x^2}\right) \quad (12).$$

This expression allows us to predict the appearance of I_S on a singly curved surface as a function of curvature. Figure 6 shows how I_S changes for a cylinder of fixed curvature as we change ϕ. It is worth noting that both the magnitude and shape of I_S change as ϕ increases.

Planes

For a planar surface, $\bar{N}$ is constant. Hence recalling our basic assumptions, I_S is constant across a plane. If the plane is oriented such that α is small enough, then a viewer will perceive elevated intensity reflected from the plane. As with the singly curved surface, the magnitude of the perceived intensity will depend on α. If α is not sufficiently small, then I_S will be zero at all points on the plane. These observations provide us with two useful pieces of information:

1. Shiny surfaces which don't generate specularities over a range of orientations are probably planar.
2. Surfaces which produce a specularity of constant intensity over a 2-D region in the image are locally planar.

5. Predicting G

In the previous analysis we have used the fact that over most viewer, source, surface configurations the geometrical attenuation factor G of (1) will have the constant value 1. For large angles of incidence, however, the character of G changes remarkably. In particular, for large angles of incidence (glancing incidence) we see that

1. G can become as large as 10.
2. G causes a shift in the peak of the specular profile toward larger angles of incidence.
3. G causes the specular profile to be unsymmetric as a function of α.

It is not surprising that when these effects are present in an image, they are rather easy to detect. For this reason, it is probably profitable to make qualitative predictions about G in applications where large angles of incidence are possible.

6. The Laboratory Setup

A laboratory arrangement has been set up to test the derived relationships (Figure 7). In this section of the paper, the laboratory setup is described. In Section 7, we share insight gained about how to best use the input image data to infer surface properties from specularities in a working system. Experimental results are presented in Section 8.

Figure 7. The Specularity Lab

To insure accurate measurements, the experiments are conducted on a 4x6 foot optical table. High precision rotation and translation stages are used to position the objects being viewed. A halogen light source with a 5 mm wide filament is placed 20 feet from the object surface to approximate a point source. Monochromatic image data is obtained using a video camera and an image digitizer. A 210 mm lens is used with the video camera to obtain high resolution across the specularity. The resulting images are in the form of 256x256 arrays of pixels. Each pixel has eight bits of gray level resolution. A precise positioning device has been built to position the camera relative to the surface. Camera-object distances of at least 24 inches are enforced to insure that the assumed distant object condition is met. Using this setup, it is possible to obtain more than 40 pixels across a specular feature which is less than a centimeter wide on the surface.

Aluminum cylinders of diameter 3.5, 2.5, 1.5, and 0.75 inches are used to test the predicted relationships (Figure 8). The cylinders have been carefully machined to achieve uniformity of surface roughness on individual cylinders and between different cylinders. The length of each cylinder is divided into 4 sections of different measured roughness. From this we are able to study the effect of varying surface roughness on images of specularities. In the future we plan to experiment with different kinds of specular surfaces and also with surfaces which are doubly curved.

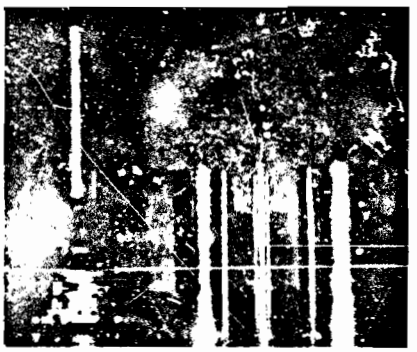


Figure 8. Experimental Specular Surfaces

7. Interpreting Real Images

In this section we discuss practical considerations related to using the results from Section 4 to infer surface properties from real images. For the first set of experiments, each cylinder is oriented such that $\alpha = 0$ on the line of maximum perceived intensity. For this special case, the doubly curved surface analysis applies to our singly curved surfaces (cylinders). Figure 9 shows a typical image obtained using this configuration. As

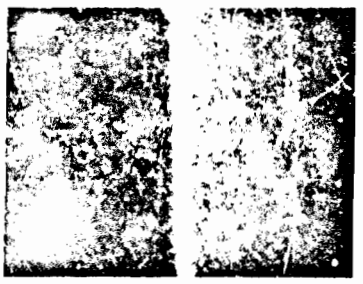


Figure 9. A Specular Image

previously discussed, the specularities are easily located in an image. It is reasonable to assume that $(I_A + I_D)$ is constant in the small neighborhood of the specularity. Thus we compute I_S by subtracting a constant from the pixel values on the specularity in an input image. I_S is set to zero elsewhere. Figure 10 shows a plot of I_S along a horizontal row of pixels taken from Figure 9 after the subtraction of $(I_A + I_D)$.

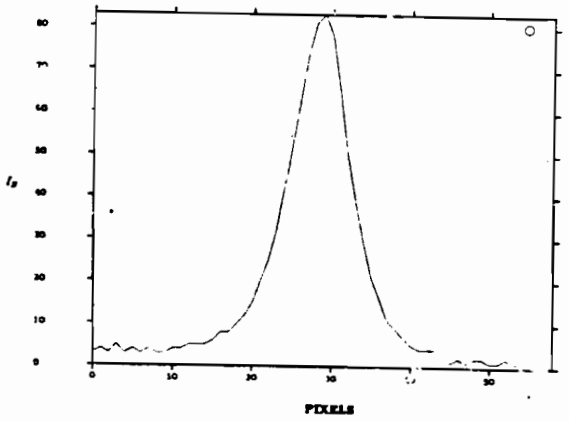


Figure 10. Plot of I_S

Computing the Curvature

Suppose now that we want to make the curvature computation described in Section 4 on the horizontal image line plotted in Figure 10. The problem is to select an appropriate pixel segment S in this image row which intersects the line of maximum I_S and begins and terminates on pixels of equal specular intensity. An equivalent problem is to draw a horizontal line in Figure 10 which intersects the I_S curve at 2 points of equal specular intensity. In principle, any image segment which satisfies the stated conditions will suffice. In practice, however, some choices for S are better than others. Figure 11 shows the case where S is chosen to be only in the high intensity part of the I_S curve. For this case S will only be a few pixels long and any error in the measurement will cause a large relative error in $|S|$. Figure 12 depicts a different kind of problem. Here S is drawn to connect two pixels of small specular intensity. Now S is many pixels long, but choosing the terminating points is difficult, if not impossible. This is because I_S changes only slightly from pixel to pixel on the fringes of the specular profile. Hence for this case there will probably be a large spatial range of pixels which are reasonable terminating points for S. Hence by choosing one of these, we risk introducing a large error in $|S|$.

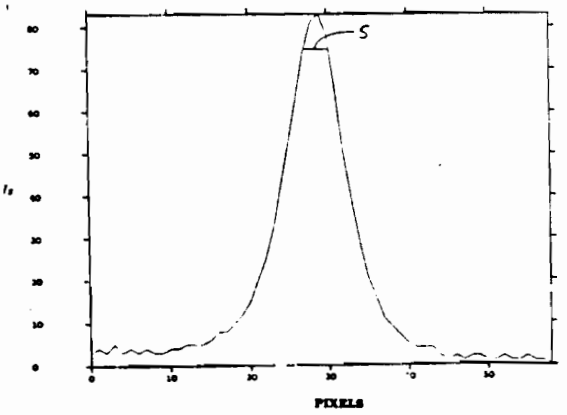


Figure 11. S Chosen Too High.

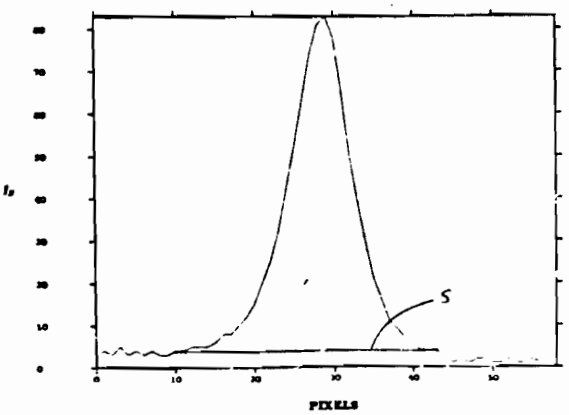


Figure 12. S Chosen Too Low

The solution, of course, is to have S begin and terminate at pixels of intermediate specular intensity. Since I_S is approximately Gaussian, we can determine a Gaussian fit for I_S and compute its standard deviation σ. Let μ be the pixel of maximum specular intensity (the mean of the Gaussian) and let T be the symmetric about μ pixel interval

$$T = [\mu - w, \mu + w] \quad (13)$$

over which S is defined. Consistent with the heuristic arguments given above, experiments have shown (see

Section 5 that an accurate curvature computation will result if we have

$$\sigma \leq w \leq 2\sigma \quad (14)$$

By imposing this restriction on T, we guarantee that we will choose an S which avoids either of the previously discussed pitfalls. We may improve our accuracy by performing the computations of Section 4 for several different S segments for which w satisfies (14). The curvature κ can then be obtained by averaging the curvatures computed for the different segments.

Truncation Effects

Since specularities are usually the brightest features in images, specular intensities are often too large to be represented in the number of bits per pixel allowed by the digitizing hardware. If this is the case, the specularity is said to be truncated. Figure 13 shows I for a truncated specularity. The obvious way to deal with this situation is to avoid it. One avoidance technique is to take multiple images in which differing amounts of light are allowed to pass through the lens. This can be achieved either by adjusting the lens aperture or by using filters. Another possible solution is to control the illumination to eliminate the possibility of truncation.

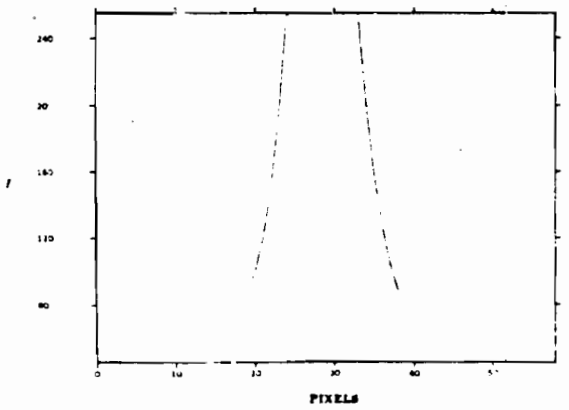


Figure 13. A Truncated Specularity

If inferences must be made from a single image, then it is arguably better to allow truncation to occur. In the case where input images have eight bits per pixel, intensities will range from 0 to 255. In many applications it is possible to weaken the incident illumination so that no truncation occurs. In doing this, however, we cause pixels on the I_S curve which previously had significant specular intensities (on the truncated specular feature) to have negligible specular intensities. The net effect of eliminating truncation is to decrease the width of the specular feature and make width measurements more susceptible to small errors.

8. Experimental Results

The technique described in Section 4 has been applied to the task of determining the curvature of each of the four cylindrical surfaces. Figure 14 shows the I profile for each of the four surfaces c_1, c_2, c_3, and c_4 taken along the line of maximum curvature. Note that we allow truncation. Table 1 displays the results of the experiments. For these measurements, w (see Section 7) is taken to be 1.85σ. We have observed that for the larger (less curved) cylinders c_1 and c_2 the computed curvature varies by less than 10% as we let w vary in the interval $\sigma \leq w \leq 2\sigma$. The reason for this desirable behavior is the fact that we are able to measure many pixels across specularities on these cylinders. On the other hand, if we attempt to measure the specular widths for the smaller cylinders c_1 and c_2 near the spec-

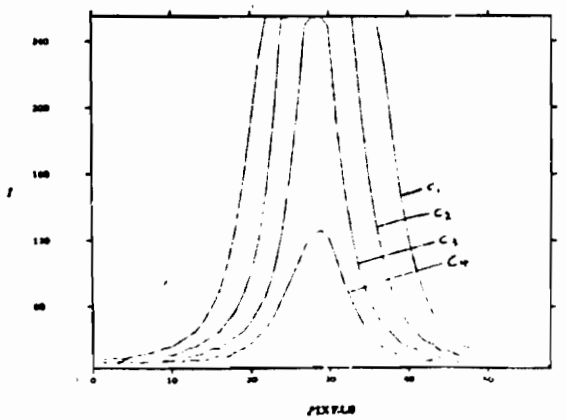


Figure 14. I for Four Different Surface Curvatures

Object	Specular Width in Pixels	κ	Computed κ	Error
c_1	24.0	0.2065	0.2065	3.4%
c_2	17.7	0.4016	0.4161	3.5%
c_3	11.4	0.6711	0.6368	5.1%
c_4	6.0	1.3514	1.2284	9.1%

Table 1. Curvature Computations for the Four Surfaces

ular intensity peaks, it is possible to get errors larger than 15% in the computed curvature. This is because near these peaks, we are only able to measure a few pixels across the specularities. Therefore for high curvature surfaces, it is advisable to measure the specular widths as near to the base of the curve as possible without failing victim to the problem discussed in Section 7. As is evident from Table 1, the more pixels which can be measured across a specular feature, the more accurately we can compute the curvature. Hence one immediate way to improve results is to digitize higher resolution images.

An interesting observation can be made from Figure 14. The specular model (2) described in Section 3, predicts that each of the four surfaces should generate the same maximum value for I_S when $\alpha = 0$. This prediction is intuitively appealing, since it seems that if we examine a small enough patch on any of the surfaces that patch should be approximately planar. But a cursory glance at Figure 13 seems to imply that a highly curved surface produces a smaller maximum value for I_S than a flatter surface. The model, however, is correct. The problem is that for the highly curved surfaces we are unable to shrink a pixel down to where the surface area it images is approximately planar. Even within the single maximum pixel, α is changing and cannot be considered to be constant zero. Hence the intensity value at the maximum pixel will be some kind of average specular intensity over a range of α and will not give us the true maximum I_S. Thus it is understandable that maximum measured intensity seems to increase as surface curvature decreases. It follows that to compute the maximum specular intensity in applications, we should use a surface of small curvature.

9. Summary and Implications

Understanding specular reflections is important for any computer vision system which must interpret images of shiny objects. Using a model developed by optics researchers, we have shown that the local orientation and principal curvatures of a specular surface can be determined by examining image intensities on a specularity. Unlike previous work, our derivations have included the effects of surface roughness and microstructure on the appearance of specular features.

A laboratory setup has been described which allows us to test our theoretical relationships. Very good results have been achieved despite the fact that the high intensity and small spatial extent of specularities make measurements difficult. Practical issues related to the implementation of our analytical results have been discussed. These issues have been addressed by the presentation of tested methods which successfully apply our results to the problem of interpreting real images.

The ability to predict intensity patch features such as specularities opens up interesting possibilities for model-based vision. Previous model-based vision systems have restricted their predictions to the shapes of image contours which will be observed for a given model. An ability to predict intensity patch features will significantly enhance the capabilities of a model-based vision system. Clearly it is advantageous to be able to make stronger predictions about an image by using additional information about the imaging process. A perhaps more important advantage of predicting intensity patches is that this prediction can provide strong guidance to low level intensity based visual processes such as edge detection. By making predictions about the appearance of intensity patch features we can hope to further unify the goals of the low level and high level mechanisms of a model based vision system.

10. Future Work

One plan for future work is to continue conducting experiments and working out details associated with inferring local surface properties from specularities. In the near future experiments are planned to study the effects of different specular materials, different kinds of roughness, and different surface orientations. We also plan to experiment with doubly curved surfaces and to develop a model which predicts the combined effects of rough surfaces and distributed light sources. It is expected that as experiments and analysis continue, we will be able to develop more refined algorithms for using specularities to understand images.

Another plan for future work is to formulate a general framework which will allow the qualitative prediction of the structure of intensity patches for a model. The primary challenge will be to isolate singularities in the structure of the intensity patches. As an example, a singularity occurs when we rotate a singly curved object from an orientation where a specularity is visible to an orientation where the specularity disappears. An ability to predict the structure and singularities of image intensity patches will open up significant new possibilities for model based vision systems.

Acknowledgements

This work has been supported by an NSF graduate fellowship, AFOSR contract F49620-82-C-0092, and ARPA contract N000-39-84-C-0211. The authors would like to thank Professor Bert Hesselink for generously providing laboratory space and equipment. We would also like to thank Rami Rise for his work in designing

the mechanical parts for the experiments.

References

[1] Blake A., "Specular Stereo," *Proceedings of IJCAI-9* (Los Angeles: August 1985), 973-976.

[2] Blinn, J., "Models of Light Reflection for Computer Synthesized Pictures," *Computer Graphics*, 11(2) (1977). 192-198.

[3] Brooks, R., "Symbolic Reasoning Among 3-D Models and 2-D Images," *Artificial Intelligence*, 17 (1981), 285-348

[4] Chelberg, D. and Lim, H. and Cowan, C., "ACRONYM Model-based Vision in the Intelligent Task Automation Project," *Proceedings of Image Understanding Workshop* (1984).

[5] Coleman, E.N. Jr. and Jain, R., "Obtaining 3-D Shape of Textured and Specular Surfaces Using Four-Source Photometry," *Computer Graphics and Image Processing*, 18 (1982), 309-328.

[6] Cook, R. and Torrance, K., "A Reflectance Model for Computer Graphics", *Computer Graphics*, 15(3) (1981), 307-316.

[7] Foley, J. and Van Dam, A., *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, 1982.

[8] Grimson, W.E.L., "Binocular Shading and Visual Surface Reconstruction," MIT AI Memo 697 (1982).

[9] Ikeuchi, K., "Determining Surface Orientations of Specular Surfaces by Using the Photometric Stereo Method", *IEEE PAMI* 3(6) (1981), 661-669.

[10] Phong, B., "Illumination for Computer Generated Pictures," *Communications of the ACM* 18 (1975), 311-317.

[11] Shafer, S., "Optical Phenomena in Computer Vision," Univ. of Rochester TR 135 (1984).

[12] Spivak, M. *Differential Geometry*, Publish or Perish, Inc., 1979.

[13] Takai, K. and Kimura, F. and Sata, T., "A Fast Visual Recognition System of Mechanical Parts by Use of Three Dimensional Model," source unknown. (First author is with CANON INC. in Tokyo.)

[14] Torrance, K. and Sparrow, E., "Theory for Off-Specular Reflection from Roughened Surfaces," *Journal of the Optical Society of America*, 57 (1967), 1105-1114.

[15] Woodham, R., "Photometric Stereo: A Reflectance Map Technique for Determining Surface Orientation from Image Intensity," Proc. SPIE, vol. 155 (1978).

GC970145-51

A Provably Convergent Algorithm for Shape from Shading

David Lee[1]

AT&T Bell Laboratories
600 Mountain Ave
Murray Hill, NJ 07974

Abstract

The problem of shape from shading and occluding boundaries is reduced to solving a system of non-linear equations by using the smoothing-spline. We present an iterative algorithm for solving this system of equations. We prove that the algorithm converges, and we analyse its complexity. We show the existence and uniqueness of the smoothing-spline and the solution of the system of equations.

1. Introduction

Research in shape from shading explores the relationship between *image brightness* and object shape. A great deal of information is contained in the image brightness values, since image brightness is related to surface orientation. Algorithms, designed to determine shape from shading, include *characteristic strip expansion* [1, 2, 3, 4, 13, 14], *photometric stereo* [5, 8, 10, 15], and *numerical shape from shading and occluding boundaries* [7].

This work is motivated by the interesting paper by Ikeuchi and Horn, [7]. That paper addresses the problem of numerical shape from shading and occluding boundaries, and results in a large system of nonlinear equations. An iterative algorithm is proposed for solving it. However, the existence and uniqueness of the solution of the system remain a problem, and the convergence of the iterative method has not been established.

We will report our preliminary work in progress on this problem. We propose using a different iterative algorithm for solving the system of nonlinear equations derived in [7], and discuss its convergence. We study the existence and the uniqueness of the solution of the system as well.

In Section 2, we describe the work by Ikeuchi and Horn briefly. In Section 3, we study a different iterative algorithm for solving the system of equations. In Section 4, we discuss the convergence of the iterative algorithm, and the existence and the uniqueness of the solution of the system of equations involved. An example for the case of a Lambertian surface is given in Section 5. In Section 6, we study the complexity of our algorithm. A different approach, from the point of view of the general theory of information-based complexity, is discussed briefly in Section 7. We conclude this paper by pointing out the limitations of this work and new directions of research.

2. Numerical Shape from Shading and Occluding Boundaries

In this section, we summarize the relevant part of [7]; the readers are referred to the original paper by Ikeuchi and Horn for details.

The goal of numerical shape from shading and occluding boundaries is to determine surface orientations from image brightness and boundary conditions. We discuss the representation of surface orientations first, and then the algorithm proposed by Ikeuchi and Horn.

2.1. Gaussian Sphere and Stereographic Projection

Surface orientation is quantified by the surface normal, a unit vector in R^3. A surface normal can be represented by a point on a unit sphere, called the *Gaussian sphere*. The part of the surface facing us corresponds to the northern hemisphere, while points on the occluding boundaries correspond to the points on the equator.

[1]This work was done when the author was at Columbia University, supported in part by an IBM graduate fellowship, in part by NSF Grant DCR-82-14322, and in part by DARPA Grant N00039-C-84-0165.

The northern hemisphere is then projected into a plane, the ξ-η plane, which is tangent to the sphere at the north pole. The projection center is the south pole. This is called *stereographic projection*. This is a conformal mapping, and the northern hemisphere is mapped onto a closed disc of radius 2 in the ξ-η plane. Therefore, points in his disc represent the surface orientations. Notice that orientations of the occluding boundaries correspond to the points on the circumferenee of that disc.

2.2. Image-Irradiance Equation and Boundary Conditions

The surface orientations are related to the image brightness by the following *image-irradiance equation*,

$$R(\xi,\eta) = E(x,y), \qquad (x,y) \in D, \tag{1}$$

where D is a unit square region, $\xi = \xi(x,y)$ and $\eta = \eta(x,y)$ represent the surface orientation, $E(x,y) \in C$ is the brightness measured at the point (x,y), and $R(\cdot\ ,\ \cdot) \in C^1$ can be determined experimentally or theoretically if some information is available about the incident, emittance and phase angles; see [5, 9]. The image-irradiance equation provides information for determining the surface orientation from image brightness.

From Equation 1 alone, one cannot determine the surface orientation (ξ,η) at each point (x,y). We need supplementary information from boundary conditions. The outline of the projection of an object in the image plane is called its *silhouette*. Some parts of it may correspond to sharp edges on the surface, and some parts to places where the surface curves around smoothly. The smooth parts of the surface correspond to the parts of the silhouette, called *occluding boundaries*, which supply important information about the shape of an object.

Without loss of generality, we assume that $R(\xi,\eta)$, $E(x,y) \geq 0$, and that $E(x,y) = 0$ iff (x,y) belongs to the occluding boundaries iff $\xi^2(x,y) + \eta^2(x,y) = 4$ iff $R(\xi(x,y),\eta(x,y)) = 0$. We further assume that the surface orientations on the boundaries of D are known.

2.3. Consistency Constraint

We assume that the surface we perceive is smooth. More specifically, we assume that the first order partial derivatives of $\xi(x,y)$ and $\eta(x,y)$ are square integrable. We also assume that real world surfaces tend to be stable, and the stability is measured by

$$\nu = \nu(\xi,\eta) = \int_D [(\xi_x)^2 + (\xi_y)^2 + (\eta_x)^2 + (\eta_y)^2]\ dxdy. \tag{2}$$

where ξ_x, ξ_y, η_x, and η_y denote the partial derivatives of ξ and η with respect to x and y, respectively. The *surface consistency constraint* is quantified as minimizing ν.

Thus, observing the image-irradiance Equation 1 and boundary information, we are seeking functions $\xi(x,y)$ and $\eta(x,y)$, which tend to minimize ν in Equation 2. An approach used in [7] is *spline-smoothing* [8]. It finds ξ and η which minimize

$$\mu = \mu(\xi,\eta) = \int_D \{ [(\xi_x)^2 + (\xi_y)^2 + (\eta_x)^2 + (\eta_y)^2] + \lambda[R(\xi,\eta) - E(x,y)]^2 \}\ dxdy. \tag{3}$$

where the *penalty parameter* λ is set according to the accuracy of the measurement of the image brightness and the preciseness of the modeling of the lighting environment by R. The noisier the measurement and the less precise the modeling, the smaller the parameter λ. For example, in [7], λ is set, heuristically, in inverse proportion to the root-mean-square of the noise in the image brightness measurements.

2.4. An Iterative Algorithm

In the previous discussion, we described the image-irradiance equation, boundary conditions, the smoothness and consistency constraint, and arrived at spline-smoothing. All quantities involved are continuous functions.

We now discretize the unit square region D in the xy-plane with mesh size h, and discretize μ by using difference operators instead of differential operators, and summations instead of integrals. The corresponding *discrete* smoothing-

spline, or DSS for short, minimizes

$$\mu = \sum_{i,j} (s_{i,j} + \lambda\, r_{i,j}). \tag{4}$$

where

$$s_{i,j} = [(\xi_{i+1,j} - \xi_{i,j})^2 + (\xi_{i,j+1} - \xi_{i,j})^2 + (\eta_{i+1,j} - \eta_{i,j})^2 + (\eta_{i,j+1} - \eta_{i,j})^2]/h^2,$$

$$r_{i,j} = [R(\xi_{i,j}, \eta_{i,j}) - E_{i,j}]^2,$$

and where $\xi_{i,j}$ and $\eta_{i,j}$ represent the surface orientation at the regular grid point (ih,jh), and $E_{i,j}$ is the brightness measured at the grid point (ih,jh). The above minimization is subject to the boundary constraints, i.e., $\xi_{i,j}$ and $\eta_{i,j}$ are known if (ih,jh) belongs to the boundaries.

To minimize μ in Equation 4, we have to solve a large system of sparse nonlinear equations:

$$\begin{aligned} \xi_{i,j} &= \xi_{i,j}^{*} - 4^{-1}\lambda h^2 [R(\xi_{i,j},\eta_{i,j}) - E_{i,j}]\, \partial R(\xi_{i,j},\eta_{i,j})/\partial \xi, \\ \eta_{i,j} &= \eta_{i,j}^{*} - 4^{-1}\lambda h^2 [R(\xi_{i,j},\eta_{i,j}) - E_{i,j}]\, \partial R(\xi_{i,j},\eta_{i,j})/\partial \eta, \end{aligned} \tag{5}$$

where

$$\xi_{i,j}^{*} = [\xi_{i+1,j} + \xi_{i,j+1} + \xi_{i-1,j} + \xi_{i,j-1}]/4, \text{ and}$$

$$\eta_{i,j}^{*} = [\eta_{i+1,j} + \eta_{i,j+1} + \eta_{i-1,j} + \eta_{i,j-1}]/4.$$

To solve Equation 5 for $\xi_{i,j}$ and $\eta_{i,j}$, Ikeuchi and Horn [7] proposed the following iterative algorithm:

$$\begin{aligned} \xi_{i,j}^{(m+1)} &= \xi_{i,j}^{*(m)} - 4^{-1}\lambda h^2 [R(\xi_{i,j}^{(m)},\eta_{i,j}^{(m)}) - E_{i,j}]\, \partial R(\xi_{i,j}^{(m)},\eta_{i,j}^{(m)})/\partial \xi, \\ \eta_{i,j}^{(m+1)} &= \eta_{i,j}^{*(m)} - 4^{-1}\lambda h^2 [R(\xi_{i,j}^{(m)},\eta_{i,j}^{(m)}) - E_{i,j}]\, \partial R(\xi_{i,j}^{(m)},\eta_{i,j}^{(m)})/\partial \eta. \end{aligned} \tag{6}$$

We can repeatedly use the values from the mth iteration on the right-hand side to compute the values for the $(m+1)$st iteration on the left-hand side. The initial values are supplied by the boundary conditions, i.e., $\xi_{i,j}$ and $\eta_{i,j}$ are known if (ih,jh) belongs to the boundaries.

The existence and uniqueness of the solution remain a problem, and the convergence of the iterative method has not been established. Furthermore, Equation 5 is a *necessary conditions* for minimizing μ in Equation 4, with the additional constraint that in a DSS, $\xi_{i,j}^{*2} + \eta_{i,j}^{*2} < 4$, if (ih,jh) does not belong to the boundary points.

3. A New iterative Algorithm

In this section, we study a different iterative algorithm for solving Equation 5, which for a range of λ converges to the unique solution of the system: the unique DSS minimizing μ in Equation 4. For an arbitrary λ, the uniqueness of the solution and the convergence of the algorithm need further study.

We first discuss a matrix which is related to the algorithm, and then we present the algorithm.

3.1. Matrix A

Let $K + 1 = n^{-1}$ and $N = K^2$. In the rest of this part, we will deal with an NxN matrix

$$A = \begin{pmatrix} B & -I & & & \\ -I & B & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & B & -I \\ & & & -I & B \end{pmatrix} \tag{7}$$

where the KxK matrix

$$B = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix} \tag{8}$$

We state a few facts about matrix A below; for the details, see [11].

Matrix A is symmetric and positive definite, with eigenvalues $\{\lambda_{i,j}\}$, i,j = 1, ..., K, where

$$\lambda_{i,j} = 4\left[\sin^2\frac{\pi i}{2(K+1)} + \sin^2\frac{\pi j}{2(K+1)}\right]. \tag{9}$$

The inverse of A, A^{-1}, is also symmetric and positive definite, with eigenvalues $\{\mu_{i,j}\}$, i,j = 1, ..., K, where

$$\mu_{i,j} = \lambda_{i,j}^{-1}. \tag{10}$$

A^{-1} can be decomposed as

$$A^{-1} = HLH, \tag{11}$$

where the diagonal matrix

$$\Lambda = \mathrm{diag}\{\mu_{i,j}\}, \quad i,j = 1, \cdots, K, \tag{12}$$

and H is the tensor product

$$H = S \otimes S, \tag{13}$$

where the (i,j)th entry of the KxK matrix S is

$$s_{i,j} = [2/(K+1)]^{1/2} \sin[\pi ij/(K+1)]. \tag{14}$$

Multiplying A^{-1} by a vector costs $O(N^2)$, using the conventional method. Since we can decompose $A^{-1} = S \otimes S \Lambda S \otimes S$ and Λ is a diagonal matrix, taking advantage of the structure of the entries of S, $s_{i,j}$, we can use Fast Fourier Transforms (FFT) for the multiplication, which costs $O(N(\log N))$.

3.2. A New Iterative Algorithm

Equation 5 can be rewritten as

$$Mx = -\lambda h^2 b(x), \tag{15}$$

where

$$M = \begin{pmatrix} A & O \\ O & A \end{pmatrix} \tag{16}$$

where A is given in Equation 7, and

$$b = \tag{17}$$

$$[\cdots, (R(\xi_{i,j}, \eta_{i,j}) - E_{i,j})\, \partial R(\xi_{i,j}, \eta_{i,j})/\partial \xi, \cdots,$$

$$(R(\xi_{i,j}, \eta_{i,j}) - E_{i,j})\, \partial R(\xi_{i,j}, \eta_{i,j})/\partial \eta, \cdots]^T,$$

and

$$x = \tag{18}$$

$$[\xi_{1,1}, \cdots, \xi_{i,K}, \cdots, \xi_{K,K},$$

$$\eta_{1,1}, \cdots, \eta_{1,K}, \cdots, \eta_{K,K}]^T.$$

Since A is non-degenerate, M is also non-degenerate, and therefore, Equation 15 is equivalent to

$$x = -\lambda h^2 M^{-1} b(x), \tag{19}$$

We propose using the following iterative algorithm

$$x^{(m+1)} = -\lambda h^2 M^{-1} b(x^{(m)}), \tag{20}$$

where $x^{(0)}$ is an arbitrary initial element.

We discuss the convergence of the iterative algorithm in Equation 20, the existence and the uniqueness of the solution of Equation 15, and the existence and the uniqueness of the DSS, which minimizes μ in Equation 4, in the next section.

4. Convergence of the Algorithm

Our goal is to find a DSS $x^* = [\xi_{1,1}^*, \cdots, \xi_{1,K}^*, \cdots, \xi_{K,K}^*, \eta_{1,1}^*, \cdots, \eta_{1,K}^*, \cdots, \eta_{K,K}^*]^T$, which minimizes μ in Equation 4, subject to the boundary constraints, i.e., the surface orientations are known on boundaries. We call a DSS *regular*, if $(\xi_{i,j}^*)^2 + (\eta_{i,j}^*)^2 < 4$, when (ih,jh) is not a boundary point. A regular DSS does not generate false occluding boundary points.

Since $(\xi_{i,j}, \eta_{i,j})$ is in the closed disc with radius 2, denoted by S, x is defined on a compact set in R^{2N}, S^{2N}. Since μ in Equation 4 is a continuous function of x, it obtains its minimum on S^{2N}, and therefore, DSS exists. We first show that a DSS is regular, and thus a regular DSS exists. We then show that the DSS is unique and that algorithm in Equation 20 converges to this unique DSS.

We assume that there exist at least two boundary points, on which the surface orientations are different.

To prove that a DSS is regular, we need

Lemma 4.1 Let P_0 be a point on the circumference of a disc, and let $P_i \neq P_0$, $i = 1,2,\ldots,k$, be points in the disc. Then for $\delta > 0$ and $K > 0$, there exists P_0^* in the disc, such that (i) $d(P_0, P_0^*) < \delta$, and (ii)

$$\sum_{i=1}^{k} d(P_0,P_i)^2 > \tag{21}$$

$$\sum_{i=1}^{k} d(P_0^*,P_i)^2 + K d(P_0,P_0^*)^2,$$

where d(P,Q) is the Euclidean distance between P and Q. □

Proof. Denote the angle spanned by the vectors P_0P_i and P_0P_j as $\langle P_i,P_0,P_j\rangle$. Let $\alpha = \max\{\langle P_i,P_0,P_j\rangle, i,j = 1,2,...,k\}$. Let $Q'P_0Q$ be the bisector of α. Since P_i is in the disc, $\alpha < \pi$, $\langle P_i,P_0,Q\rangle < \pi/2$, and $\langle P_i,P_0,Q'\rangle = \pi - \langle P_i,P_0,Q\rangle > \pi/2$. Let P_0^* be a point on P_0Q such that $d(P_0,P_0^*) < \delta$. Then in the triangle $P_0P_0^*P_i$, $d(P_0,P_i)^2 = d(P_0^*,P_i)^2 + d(P_0,P_0^*)^2 - 2d(P_0^*,P_i)\, d(P_0,P_0^*) \cos\langle P_0,P_0^*,P_i\rangle$. Since $\lim_{P_0^* \to P_0} \langle P_0,P_0^*,P_i\rangle = \langle P_i,P_0,Q'\rangle > \pi/2$, and $\lim_{P_0^* \to P_0} P_iP_0^* = P_iP_0$, then for sufficiently small $d(P_0,P_0^*)$, it is seen that $-2d(P_0^*,P_i) \cos\langle P_0,P_0^*,P_i\rangle > Kd(P_0,P_0^*)$. Therefore, $-2d(P_0^*,P_i)\, d(P_0,P_0^*) \cos\langle P_0,P_0^*,P_i\rangle > Kd(P_0,P_0^*)^2$, and $d(P_0,P_i)^2 > d(P_0^*,P_i)^2 + Kd(P_0,P_0^*)^2$. Taking summation over all i, we have Inequality 21. □

We are ready to prove

Lemma 4.2 A DSS is regular. □

Proof. We prove by contradiction. We assume, on the contrary, that there exists a DSS x^*, which is not regular, i.e.,

$$\mu(x^*) = \sum_{i,j} (s_{i,j}^* + \lambda\, r_{i,j}^*), \tag{22}$$

where

$$s_{i,j}^* = [(\xi_{i+1,j}^* - \xi_{i,j}^*)^2 + (\xi_{i,j+1}^* - \xi_{i,j}^*)^2 + (\eta_{i+1,j}^* - \eta_{i,j}^*)^2 + (\eta_{i,j+1}^* - \eta_{i,j}^*)^2]/h^2,$$

$$r_{i,j}^* = [R(\xi_{i,j}^*, \eta_{i,j}^*) - E_{i,j}]^2,$$

and there exists $(\xi_{i,j}^*, \eta_{i,j}^*)$, such that $\xi_{i,j}^{*2} + \eta_{i,j}^{*2} = 4$. Since (ih,jh) is an interior point of the region D, $E_{i,j} > 0$.

We have

$$\mu(x^*) = F(x^*) + \lambda[R(\xi_{i,j}^*,\eta_{i,j}^*) - E_{i,j}]^2 + \{[(\xi_{i+1,j}^* - \xi_{i,j}^*)^2 + (\eta_{i+1,j}^* - \eta_{i,j}^*)^2] + [(\xi_{i,j+1}^* - \xi_{i,j}^*)^2 + (\eta_{i,j+1}^* - \eta_{i,j}^*)^2] + [(\xi_{i-1,j}^* - \xi_{i,j}^*)^2 + (\eta_{i-1,j}^* - \eta_{i,j}^*)^2] + [(\xi_{i,j-1}^* - \xi_{i,j}^*)^2 + (\eta_{i,j-1}^* - \eta_{i,j-1}^*)^2]\}/h^2, \tag{23}$$

where $F(x^*)$ does not contain $\xi_{i,j}^*$ and $\eta_{i,j}^*$.

Let $d(k,l;i,j)$ be the Euclidean distance between $(\xi_{k,l}^*, \eta_{k,l}^*)$ and $(\xi_{i,j}^*, \eta_{i,j}^*)$. Then

$$\mu(x^*) = F(x^*) + \lambda[R(\xi_{i,j}^*, \eta_{i,j}^*) - E_{i,j}]^2 + [d^2(i+1,j;i,j) + d^2(i,j+1;i,j) + d^2(i-1,j;i,j) + d^2(i,j-1;i,j)]/h^2. \tag{24}$$

Since $E_{i,j} > 0$, $R(\xi_{i,j}^*, \eta_{i,j}^*) = 0$ and R is continuous, there exists a disc, centered at $(\xi_{i,j}^*, \eta_{i,j}^*)$ with sufficiently small radius δ, such that for all (ξ,η) inside the disc, $[R(\xi,\eta) - E_{i,j}]^2 \le [R(\xi_{i,j}^*, \eta_{i,j}^*) - E_{i,j}]^2 = E_{i,j}^2$.

Let $P = \{(i+1,j), (i-1,j), (i,j+1), (i,j-1)\}$, and let $A = \{(k,l) \in P:\ d(k,l;i,j) > 0\}$ and $K = |P - A|$. We analyze the following two cases, and arrive at a contradiction for each case.

Case 1. $K < 4$, i.e., there exists at least one (k,l), such that $d(k,l;i,j) > 0$. By applying Lemma 4.1, with $P_0 = (\xi_{i,j}^*, \eta_{i,j}^*)$ and $\{P_i\} = A$, we know that there exists (ξ^*,η^*), inside the disc, centered at $(\xi_{i,j}^*, \eta_{i,j}^*)$, with radius δ, such that

$$\sum_{(k,l)\in P} d(k,l;i,j)^2 = \sum_{(k,l)\in A} d(k,l;i,j)^2$$

$$> \sum_{(k,l)\in A} d^*(k,l;0,0)^2 + Kd^*(i,j;0,0)^2$$

$$= \sum_{(k,l)\in A} d^*(k,l;0,0)^2 + \sum_{(k,l)\in P-A} d^*(k,l;0,0)^2$$

$$= \sum_{(k,l)\in P} d^*(k,l;0,0)^2,$$

where $d^*(k,l;0,0)$ is the Euclidean distance between $(\xi_{k,l}^*, \eta_{k,l}^*)$ and (ξ^*,η^*).

Replacing $(\xi_{i,j}^*, \eta_{i,j}^*)$ by (ξ^*,η^*), from Equation 24, we have x^{**}, such that $\mu(x^{**}) < \mu(x^*)$. Therefore, x^* is not a DSS, a contradiction.

Case 2. $K = 4$, i.e, $d(k,l;i,j) = 0$, $\forall\ k,l \in P$. We separate out all the grid points, adjacent to any of the points in P. We denote this set of grid points by $P^{(1)}$. We analyze $P^{(1)}$, in a similar way as P, and we will arrive either at a contradiction or at the conclusion that $d(k,l;i,j) = 0$, $\forall\ (k,l) \in P \cup P^{(1)}$, i.e., $(\xi_{k,l}^{\ *}\ ,\ \eta_{k,l}^{\ *}) = (\xi^*,\eta^*)$, $\forall\ (k,l) \in P \cup P^{(1)}$. We repeat the same arguments as we expand the region of grid points with identical (ξ^*,η^*). Since D is a unit square region and there exist at least two boundary points on which the (fixed) surface orientations are different, we will arrive at a contradiction no later than that the expanded region covers these two points.

Therefore, an irregular DSS does not exist. This completes the proof. []

We assume that $\{R(\xi,\eta) - E_{i,j}\}\partial R(\xi,\eta)/\partial\xi$ and $\{R(\xi,\eta) - E_{i,j}\}\partial R(\xi,\eta)/\partial\eta$ are *Lipschitz* functions, i.e.,

$$|\{R(\xi,\eta)-E_{i,j}\}\partial R(\xi,\eta)/\partial\xi - \{R(\xi',\eta')-E_{i,j}\}\partial R(\xi',\eta')/\partial\xi'|$$

$$\leq L^{(1)}_{i,j}\{(\xi - \xi')^2 + (\eta - \eta')^2\}^{1/2}$$

and (25)

$$|\{R(\xi,\eta)-E_{i,j}\}\partial R(\xi,\eta)/\partial\eta - \{R(\xi',\eta')-E_{i,j}\}\partial R(\xi',\eta')/\partial\eta'|$$

$$\leq L^{(2)}_{i,j}\{(\xi - \xi')^2 + (\eta - \eta')^2\}^{1/2}.$$

Let $\max\{L_{i,j}^{(k)}\}_{i,j,k} = \nu_0$. Then we have

Theorem 4.1 For $\lambda \in [0\ ,\ 2\pi^2\nu_c^{-1}|1-\pi^2h^2/24|^2)$, there exists a unique DSS minimizing μ in Equation 4, which is also the unique solution of Equation 15, and the algorithm in Equation 20 converges to that DSS. []

Proof. Since DSS exists and is regular. Equation 15 is a necessary condition of a DSS. Let x^* be a regular DSS. Then $x^* = -\lambda h^2M^{-1}b(x^*)$. From Equation 20 we have $x^{(m+1)} - x^* = -\lambda h^2M^{-1}[b(x^{(m)}) - b(x^*)]$, and so $\|x^{(m+1)} - x^*\|_2 \leq \lambda h^2\|M^{-1}\|_2\nu_0\|x^{(m)} - x^*\|_2$. Since $\|M^{-1}\|_2 = [8\sin^2(\pi h/2)]^{-1} < |2\pi^2h^2(1 - \pi^2h^2/24)^2]^{-1}$, $\|x^{(m+1)} - x^*\|_2 \leq \lambda h^2[2\pi^2h^2(1 - \pi^2h^2/24)^2]^{-1}\nu_0\|x^{(m)} - x^*\|_2 = \lambda[2\pi^2(1 - \pi^2h^2/24)^2]^{-1}\nu_0\|x^{(m)} - x^*\|_2$. Since $\lambda < 2\pi^2\nu_0^{-1}(1 - \pi^2h^2/24)^2$, $\lambda[2\pi^2(1 - \pi^2h^2/24)^2]^{-1}\nu_0 < 1$. Therefore, $x^{(m)}$ converges to x^*.

Since $x^{(m)}$ has only one limit and Equation 15 is a necessary condition satisfied by a regular DSS, $x^{(m)}$ converges to the unique solution of Equaiton 15, which is the unique DSS. []

5. An Example

As an example, we estimate ν_0 and the range of λ for the case of a Lambertian surface with the incident rays coincident with the view direction. In this case, the image-irradiance equation [7]

$$R(\xi,\eta) = (4 - \xi^2 - \eta^2)/(4 + \xi^2 + \eta^2), \qquad (26)$$

where $\xi^2 + \eta^2 \leq 4$. Obviously, R is a Lipschitz function and

$$\nu_0 \leq \qquad (27)$$

$$\{\sup|\partial/\partial\xi\{[R(\xi,\eta) - E_{i,j}]\partial R(\xi,\eta)/\partial\xi\}|^2 +$$

$$\sup|\partial/\partial\eta\{[R(\xi,\eta) - E_{i,j}]\partial R(\xi,\eta)/\partial\eta\}|^2\}^{1/2}.$$

Since $|\partial/\partial\xi\ \{[R(\xi,\eta) - E_{i,j}]\partial R(\xi,\eta)/\partial\xi\}| = |[\partial R/\partial\xi]^2 + (R - E_{i,j})\partial^2R(\xi,\eta)/\partial\xi^2| \leq |\partial R/\partial\xi|^2 + |R - E_{i,j}|\ |\partial^2R(\xi,\eta)/\partial\xi^2|$, we only need to estimate the bounds of the absolute values of $\partial R/\partial\xi$ and $\partial^2R(\xi,\eta)/\partial\xi^2$.

Since $\partial R/\partial\xi = -16\xi/(4 + \xi^2 + \eta^2)^2$. One checks that $|\partial R/\partial\xi| \leq 3\sqrt{3}/8$.

On the other hand, $\partial^2R/\partial\xi^2 = 16(3\xi^2 - \eta^2 - 4)/(\xi^2 + \eta^2 + 4)^3$. One checks that $|\partial^2R/\partial\xi^2| \leq 1/4$.

Since $0 \leq R, E_{i,j} \leq 1$, $|R - E_{i,j}| \leq 1$. Thus $|\partial/\partial\xi\ \{[R(\xi,\eta) - E_{i,j}]\partial R(\xi,\eta)/\partial\xi\}| \leq 27/64 + 1/4 = 43/64$.

A similar analysis yields $|\partial/\partial\eta\ \{[R(\xi,\eta) - E_{i,j}]\partial R(\xi,\eta)/\partial\eta\}| \leq 27/64 + 1/4 = 43/64$.

From Equation 27,

$$\nu_0 \leq 43\sqrt{2}/64. \qquad (28)$$

Thus for $\lambda \in [0\ ,\ 64\sqrt{2}\pi^2(1-\pi^2h^2/24)^2/43]$, the algorithm in Equation 20 converges to the unique DSS.

6. Complexity of the Iterative Algorithm

From Subsection 3.1 and Equation 16, M^{-1} is known. Therefore to implement the algorithm in Equation20, one has to multiply the 2Nx2N dense matrix M^{-1} by a vector, which costs $O(N^2)$ using the conventional matrix multiplication. however, as discussed in Subsection 3.1, we can use FFT to reduce the cost to $O(N(\log N))$.

Let x^* be the unique solution of Equation 15 and let $\lambda\nu_0[2\tau^2(1-\pi^2h^2/24)^2]^{-1} = \theta < 1$. Then by a similar argument as in the proof of Theorem 4.1, we have $\|x^{(m)} - x^*\|_2 \leq \theta \|x^{(m-1)} - x^*\|_2$, and therefore,

$$\|x^{(m)} - x^*\|_2 \leq \theta^m \|x^{(0)} - x^*\|_2 . \qquad (29)$$

As an example, we derive the number of iterative steps to compute a solution of Equation 15 with error bound $O(h)$. Let k be the number of steps required, then we have

$\theta^k = h$,

$k = \log h / \log \theta$.

If $\theta = 1/2$, then $k = (\log N)/2$. Thus

Proposition 6.1 For $\lambda \in [0 , \pi^2\nu_0^{-1}[1-\pi^2h^2/24]^2)$, it takes $(\log N)/2$ steps for $x^{(m)}$ to converge to the solution of Equation 15 with error h. The total cost, using FFT for matrix multiplication, is thus $O(N(\log N)^2)$.

7. Interpolating Spline and its Optimality

When the data are noisy, the spline-smoothing approach is appropriate. However, when the data are relatively precise, the *interpolating spline* approach is preferable. In that approach, one seeks a spline that interpolates the data and minimizes Equation 2. This approach is also briefly discussed in [7]. This is an *interpolatory algorithm* and is therefore *almost strongly optimal*, i.e., strongly optimal within a factor of 2, see [12], Chapter 1. The uniqueness of the spline and its construction need further investigation.

8. Conclusion

We studied an algorithm for solving the system of equations for shape from shading, using spline-smoothing, and we discussed its convergence and complexity. We proved the existence and the uniqueness of the smoothing-spline and the system of equations involved. However, the work is far from being completed. There are a number of aspects which deserve further investigation.

The image domain we use is a square region, and from the practical point of view, it is too restrictive. An efficient algorithm for general image domains remains to be explored. We assume that the surface orientations are known on the image boundaries. This is not always the case, because of sharp edges on the boundaries or noise in the data. Finding a robust algorithm with noisy or incomplete information on the boundaries is an interesting research problem. We studied the algorithm under the assumption that the penalty parameter λ is within a bound. For arbitrary λ, the existence and the uniqueness of the solution of the system of equations involved and the convergence of the algorithm need further study. Finally, our algorithm remains to be tested to characterize its performance on real data.

Acknowledgements

The author is grateful to J. R Kender, G. W. Wasilkowski, and H. Wozniakowski for their advice during this work. The valuable comments from H. Trickey are deeply appreciated.

References

1. Bruss, A. R. "Some Properties of Discontinuities in the Image Irradiance Equation." *AI Memo, AI Lab. MIT, 517* (1979).

2. Horn, B.K.P. "Shape from Shading: a Method for Obtaining the Shape of a Smooth Opaque Object from One View." *TechnicalReportMAC-TR-79* , (Project MAC, MIT 1970), .

3. Horn, B.K.P. Determining Shape from Shading. In *The Psychology of Computer Vision*, Winston, P.H., Ed.,McGraw-Hill, New York, 1975, ch. 4.

4. Horn, B.K.P. "Understanding Image Intensities." *Artificial Intelligence 8*, 2 (April 1977), 201-231.

5. Horn, B.K.P., Woodham, R.J. and Silver, W.M. "Determing Shape and reflectance Using Multiple Images." *AIMemo* , AI Lab. MIT 490 (1978), .

6. Horn, B.K.P., and Sjoberg, R.W. "Calculating the Reflectance Map " *Applied Optics 18* (June 1979), 1770-1779.

7. Ikeuchi, K. and Horn, B. K.P. "Numerical Shape from Shading and Occluding Boundaries." *Artificial Intelligence 17* (1981), 141-184.

8. Laurent, P. J. *Approximation et optimisation.* Hermann, Paris, 1972.

9. Nicodemus, F. E., Richmond, J. C., Hsia, J. J., Ginsberg, I. W., and Limperis, T. "Geometrical Considerations and Nomenclature for Reflectance." *NBS monograph 160 US Department of Commerce* (1977), National Bureau of Standards.

10. Silver, W. "Determining Shape and Reflectance Using Multiple Images." *Maser Th. Dept. of E. E. and CS* (MIT 1980).

11. Smith, G. D.. *Numerical Solution of Partial Differential Equations: Finite Difference Methods.* Oxford University Press, 1978.

12. Traub, J. F., and Wozniakowski, H. *A General Theory of Optimal Algorithms.* Academic Press, 1980.

13. Woodham, R.J. A Cooperative Algorithm for Determining Surface Orientations from a Single View. Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Aug., 1977, pp. 635-641.

14. Woodham, R.J. *Reflectance Map Techniques for Analyzing Defects in Metal Castings.* Ph.D. Th., MIT Artificial Intelligence Lab., June 1978. Available as AI-TR-457

15. Woodham, R.J. "Photometric Method for Determining Surface Orientation from Multiple Images." *OpticalEngineering 19(1)* (1980), 139-144.

GC970145-52

GENERALIZED CONE DESCRIPTIONS FROM SPARSE 3-D DATA[1]

Kashipati G. Rao and R. Nevatia

Intelligent Systems Group
Departments of Electrical Engineering
and Computer Science
Powell Hall Room 234
University of Southern California
Los Angeles, CA 90089-0273

ABSTRACT

This paper presents an approach to describing objects as Generalized Cones (GCs) starting from sparse 3-D data, such as that obtained from stereo. The current method is best suited to Linear Straight Homogeneous GCs [1], though we believe it could be extended to more complex GCs. The method has been tested on a number of synthetic images and results for some are presented.

1. INTRODUCTION

Segmentation of a scene into objects, or the resolution of the so called "figure-ground" problem is one of the key problems in computer vision. This problem has been solved for polyhedral scenes if perfect line drawings are given as input. Methods available for more general objects are highly restricted so far. The usual approach to cope with the imperfections in the low-level descriptions is to assume that the specific objects to be viewed, and even their approximate orientations, are known a priori, and then segmentation is performed by fitting the models to the low-level descriptions. The various systems differ in the specificity with which the object models must be known.

In this paper we propose an approach to segmenting scenes assuming that the objects to be viewed are well described as generalized cones. The current method is best suited to "linear, straight, homogeneous" generalized cones (in Shafer's terminology [1]), though we believe that the method generalizes to a much broader class of objects. We assume that our low-level descriptions consist of sparse 3-D data, as might be generated by a stereo system, for example. Thus, information is available only at the intensity discontinuities, typically at the object boundaries, surface discontinuities and surface markings. We do not assume that the 3-D data is available everywhere on the boundary, i.e., we must reason with incomplete, and imperfect data.

[1]This research was supported, in part, by the Defense Advanced Research Projects Agency and was monitored by the Air Force Wright Aeronautical Laboratories under contract F33615-84-K-1404, Darpa order no. 3119

2. A REVIEW OF GENERALIZED CONE METHODS

We will not attempt a very detailed or complete description of the work on generalized cones here; a tutorial treatment may be found in [2]. Generalized cones were introduced by Binford as useful volume descriptions for 3-D objects [3]. A Generalized Cone (GC) consists of an arbitrary planar shape, called a *cross-section* swept along an arbitrary 3-D curve, called an *axis*. Further, the size and also shape of the cross-section may change along the axis; the rule describing the change is called the *cross-section function*.

Thus, a normal cylinder consists of a circular cross-section, swept along a straight axis with no change in shape or size of the cross-section. For a normal cone, the size changes linearly along the axis. The precise restrictions for a GC have been different for various researchers; e.g. some do not allow the cross-section shape to change. Shafer has developed a terminology for describing the variants of a generalized cone [1]; we shall follow this terminology where appropriate.

In this terminology a *linear* cone has a linear cross-section function, a *straight* cone has a straight axis, and a *homogeneous* cone has a invariant cross-section shape. We shall exploit some special properties of a Linear, Straight, Homogeneous Generalized Cone (LSHGC).

While it has been generally accepted that GCs have many advantages as a representation of shape, it is difficult to derive these descriptions from the scenes. In previous work, Nevatia and Binford [4] used the boundaries derived from 3-D range data; the method would apply to 2-D boundaries also, but in either case requires that boundaries be complete. Marr describes another method for determining axes, also from complete boundaries [5]. Brooks deals with imperfect data in his ACRONYM system [6], but this requires rather detailed knowledge of the object being viewed, and some knowledge of the viewing position.

Deriving GC descriptions from a monocular image is a complex task. If we are able to gather 3-D data, from an active range finder, or stereo, some of the difficulties of segmentation disappear. In practice, however, it is difficult to get perfect range data everywhere. For stereo analysis in particular, it is possible to get range data for only the non-homogeneous parts of the image. Thus, for smooth objects, we may be able to get this information at object boundaries only. Depth at other points may be determined by interpolation (e.g. see [7, 8]); however, good interpolation requires the knowledge of surface discontinuities itself, i.e., the solution of the object segmentation problem. (In fact, we were motivated to this problem in trying to devise an interpolation scheme for our stereo program.) Even for an active range finder, there may be many spots where range data cannot be reliably obtained, e.g. due to color or the reflectivity properties of the surface at these points.

3. OUR APPROACH

We assume that we have sparse 3-D data for a scene, as may be expected from a line or edge based stereo system (such as [9]); i.e., we have 3-D data available at the points that are detected as edges and are found to have a corresponding match in the stereo pair.

The boundaries in any scene may be considered to be in the following classes (see figure 1 for an example):

1. Occluding boundaries:
   This is a boundary where the visible surface is all to one side of the boundary (shown by "1" in figure 1). Previous contour analysis, of Nevatia and Binford, and of Marr, essentially assumes that these are the only visible contours.

2. Surface slope discontinuities:
   These may be caused by slope discontinuities in the cross-section shape, such as along the corners of the cross-sections of a polyhedral object or by a "terminator" (i.e. the end) of a GC (The slope discontinuities are shown by "2" in figure 1). These kinds of boundaries would cause difficulties in the earlier methods of analysis, but we will show how they may provide very valuable pieces of information.

3. Surface Markings:
   These are caused by changes in the surface reflectance rather than the surface position or slope ("3" in figure 1). In simplistic analysis, these may be confused with occluding boundaries.

4. Others:
   Other sources are due to noise, shadows, highlights etc. Our approach does not deal with them explicitly, but should work in their presence (we can essentially consider them to be same as surface markings for our analysis).

1. Occluding boundaries
2. Surface orientation discontinuities
3. Surface markings

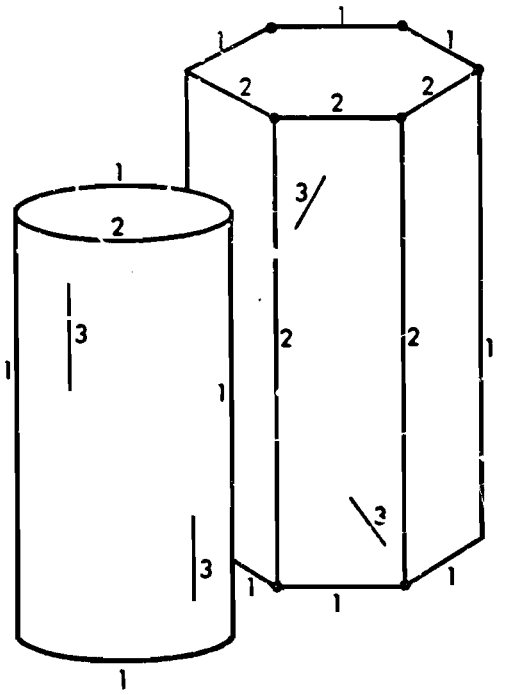


Figure 1: Two Occluding Objects And A Classification Of The Boundaries

For GCs, the important boundaries could be alternatively classified as those produced from "contour generators (cgs)" and "terminators". Intuitively, contour generators are the extremal points on the surface which enclose the visible surface (and are thus view-point dependent). For a smooth GC, the contour generators are the points on the surface where the line of sight is tangential to the viewed surface. More generally, the contour generators are the loci of the extremal points on the cross-sections. (In this, our definition is slightly different from that used in Shafer, but has the same intuitive notion. Also we will not distinguish between "contours" which are projections in the image of contour generators and contour generators themselves, unless necessary to do so, as we use 3-D boundaries.) Contour generators for a simple cylinder are shown in figure 2. Note that in our terminology, contour generators are a part of the occluding boundaries. In an SHGC we usually have 2 segments for the contour generator separated by the terminators. These shall be referred to as cg1 and cg2.

The terminators of a GC are simply its ends (imagine an infinite GC cut at a point). Note that the cut, and hence a terminator need not be planar, and when planar, it

need not be normal to the axis. (Thus, we prefer to describe a right, circular cone with a slanted cut, as a straight, homogeneous GC with an oblique termination, rather than as a homogeneous, GC with cross-section changing shape at the end, i.e. our descriptions are necessarily in terms of right GCs.) Terminator boundaries for a simple cylinder are shown in figure 2; we may note that the terminator may share part of the occluding boundary. The terminators have been a source of difficulty in analysis of boundaries, as in [4]; however, they can provide valuable clues to the shapes of the cross-sections.

CG: Contour Generator
T: Terminator Boundary

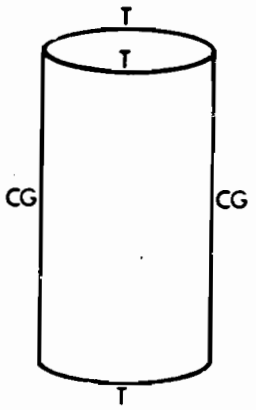


Figure 2: Another Classification For CC Boundaries

The scene segmentation problem may, now, be considered to be that of isolating the contour generators and terminators of the GCs present in a scene. Also, having these boundaries goes a long way towards describing the GCs, the axis comes from the axis of symmetry of the contour generators and the cross-section shape comes from the terminators under certain conditions.

The key to our approach consists of the following observations about boundaries of GCs:

a) The contour generator is tangential (in 3-D) to the terminator boundaries and 3-D tangency also implies a 2-D tangency. Further, the contour generator (and thus the entire object) must be to one side of the plane containing the local contour tangent and the viewing point. (In 2-D, the terminator boundary must be all on one side of the contour at the junction.)

b) For a linear, straight, homogeneous GC (LSHGC), the contour generator is planar from any view (established by Shafer [1]).

For a non-linear SHGC, the contour generators are planar in a side view, but not necessarily in an oblique view. Properties of contour generators of unconstrained GCs are not known, but we expect that they can be approximated by piecewise LSHGCs, giving rise to "piecewise coplanar" contour generators. In our current implementation, we have tested only the LSHGCs, but believe that the methods will extend to elongated GCs by using the piecewise approximation.

Our approach, then, is to find terminators and contour generators by testing for the above properties. In the current implementation we search for all alternatives and then choose the preferred descriptions based on certain preference criteria. In general, the search itself will need to be constrained for more complex scenes. The details of the method and a discussion follow.

4. DETAILS OF THE METHOD

The input to this system is a set of 3-D line segments, the output is a set of descriptions of objects using the GC representation.

The block diagram of the method is given in figure 3, details are in the following sub-sections. As an example, we shall consider figure 4, which is a projection of hypothetical 3-D sparse data for a scene with a cylinder. We assume the 3-D segments 1 and 6 lie in the plane of the figure.

4.1. Preliminary Processing

• Find Lines: Line segment information comes from a lower level program and consists of the positions of end-points of each line in 2-D and in 3-D. Other attributes, such as length, are computed.

• Find Line Relations: Relationships between every pair of line segments like coplanarity, parallelism and convergence (meeting at a junction) are found.

• Find Junctions: A junction is formed when several segments meet at a point or when the distance between their extremities is within a preset threshold.

4.2. Searching For Possible GCs

We next search for groups of line segments that can be looked upon as GCs. Our approach consists of searching for some evidence of existence of a GC and then verifying it with other evidence. Our search can proceed in one of the following two modes:

1. Find contour generators first and verify by finding the corresponding terminator(s) (called the Contour Generator Directed Method).

2. Find the terminator(s) first and verify by finding the corresponding contour generator(s) (called the Terminator Directed Method).

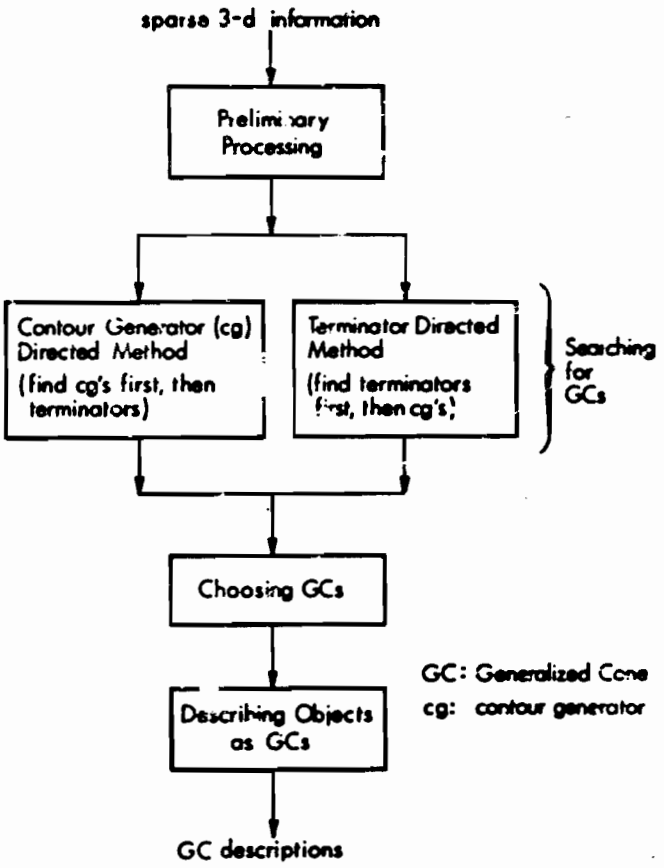


Figure 3: Block Diagram Of The Method

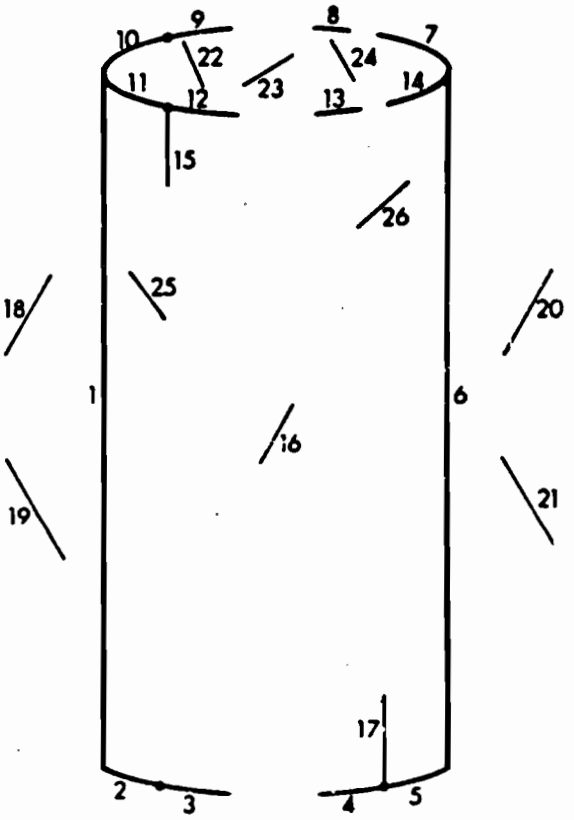


Figure 4: Illustrating The Method With A Cylinder

These two methods are described in detail below. At this time, we use both methods to give a list of possible GCs and choose among them by some preference criteria (given later). A complete search can be expensive for a complex scene and will need to be guided by results of a partial analysis. We have not investigated the issues of controlling such a search yet; our approach will be that the "strongest" evidence is to be used first, and that, generally, sufficient strong evidence can be found that exhaustive search is unnecessary.

4.2.1. The Contour Generator Directed Method

Each pair of coplanar line segments is a possible pair of contour generators. To prune the search space, we consider only those segments whose length is greater than the average-segment-length (average length of segments in the scene). For each such pair, we look for the corresponding terminator(s), by the following method:

- Find those segments which, when projected to the contour generator plane, lie in between the contour generators. Only such segments could belong to the terminators, as the cgs are, by definition, the extremities of the object from a certain point of view. In figure 4, if we consider (1, 6) as possible cgs then all but segments 18, 19, 20 and 21 would be in between the cgs and could belong to the terminators of (1, 6).

- Find begin and end segments of terminators: For each end point of each contour generator and for each side of the contour generator plane, pick the segment, one of whose extremities is closest to the cg end point. This is justified because on each side of the cg plane there is a unique begin/end segment associated with each end point of a contour generator. Note that we use this method rather than connectivity because, in general, the junction between the cg and the terminator may be missing. Segments associated with one of the contour generators are called begin segments and those associated with the other cg are called end segments. In figure 4, segment 11 is chosen as a begin segment rather than segment 25 because 11 is closer to the extremity of the contour generator segment 1. The other begin segments found are 10 and 2; the end segments found are 7, 14 and 5.

- Starting from each begin segment trace the possible terminator till an end segment is reached, using the algorithm for tracing a contour explained later, in paragraph 4.2.3. For example, in figure 4 we get (2, 3, 4, 5) as a possible terminator.

4.2.2. The Terminator Directed Method

We find ordered sets of segments that are potential terminators by the following method:

- Find maximal coplanar sets, i.e., the largest set of segments that are all in the same plane (we consider sets with 3 or more segments). For noisy data, we find the set of segments lying inside a thin slab rather than those lying strictly in a plane. In figure 4 segments (7, 8, 9, 10, 11, 12, 13, 14, 22, 23, 24) would form one maximal coplanar set.

- Find maximal convex sets: For each coplanar set found above, we remove those segments that do not lie on the largest convex hull. A segment is removed if there are segments lying on both of its sides. For example, 22, 23 and 24 are removed from the maximal coplanar set found above, to obtain (7, 8, 9, 10, 11, 12, 13, 14). At this point, the segments in the set are not ordered.

- Find the extremal segments: the extremal segments belong to the two sections of the contour with minimum angle in the cg plane (this is the linear-segment-version of obtaining the sections of maximum/minimum curvature of a curve). Note that, as in an inclined cylinder, we cannot simply consider the leftmost and the rightmost segments as the extremal segments of the terminator. We now obtain a pair of extremal segments per section and, consequently, four for each terminator. One pair is called the begin segments and the other pair is called the end segments, for convenience in tracing the terminator (described below). Segments 11, 10 are the begin segments and 7, 14 are the end segments of the maximal convex set obtained above.

- For each set of segments found above, we look for the corresponding contour generators, i.e., coplanar segments that do not lie in the terminator plane and are closest to the extremal segments of the terminator. Segments 1 and 6 satisfy this for the above set. We then trace the terminator by starting from each begin segment and taking only those segments that are in the maximal convex set, until an end segment is reached. For this we use the tracing procedure explained next. In the example, the terminator we obtain is (11, 12, 13, 14, 7, 8, 9, 10) ordered from cg=1 to cg=6.

4.2.3. Tracing A Contour

Both of the above methods need a technique to trace the terminator. The tracing problem consists of starting from a begin segment and linking it to its neighbors in a chain until an end segment is reached, such that some constraints are satisfied.[2]

This is in general a difficult problem because of the presence of surface markings and other noise segments. We may also have gaps between segments and hence cannot always use connectivity. Also the shape of the contour is not known a priori, hence standard transformation techniques like the Hough transform are not applicable.

[2] When two chains have their begin and end segments meeting, they are coalesced to form a single contour. In the current example, chains (11, 12, 13, 14) and (10, 9, 8, 7) are combined to form the contour (11, 12, 13, 14, 7, 8, 9, 10).

In our method, to trace the contour, we first use connectivity in the direction of traversal, i.e., we look for the next segment as the one connected to the current segment but away from the previous one. For example, in figure 4 if the current segment is 11 and the previous segment is 1 (the cg), the next segment is either 12 or 15.

Choosing From Among Several Segments: At a junction, we need to choose among many alternative branches. For example, in figure 4, starting from segment 11, we could choose either 12 or 15 on the basis of connectivity alone. This problem arises especially when we have surface markings meeting the terminator as illustrated in the figure. We would like to ignore the surface mark (segment 15) and instead continue tracing the terminator (take segment 12).

We choose the segment most contiguous with the current segment, i.e., the segment most collinear with the current one. In the example, 12 is more collinear with 11 than is 15 with 11. We, therefore, choose 12.

Crossing Gaps: Sometimes, however, we may not have any segment connected to the current segment in the direction of traversal, such as at the end of segment 12 in figure 4. To choose the most appropriate next segment, we search in a neighborhood of the current segment for segments satisfying the following constraints and preferences:

basic constraints

* The next segment should be in the general direction of traversal so far. (It should be closer to the current segment than to the previous one and not have a junction with the previous segment.) For example, in our current path (11 and 12), segment 15 does not qualify but segments 23, 16 and 13, among other segments, do.

* The next segment should not be a part of the terminator traced so far. Segment 11 does not qualify as it is already part of the terminator.

additional constraints

(applicable only if we have already found contour generators)

* The next segment should lie in between the 2 contour generators being considered. For example, segments 20, 21, 18 and 19 would not qualify.

* The next segment should lie on the same end of the contour generators as the current and previous segments. Two segments lie on the same end of the cgs if they lie on the same side (in the cg plane) of the straight line through the midpoints of the cgs. In this example, the next segment and segments 11, 12 should lie on the same end of the cgs 1 and 6. Thus segments 4, 17, 5 etc. do not qualify as the next segment but 13, 14, 8, 7, 24 and 26 do.

* The next segment should also lie on the same side of the terminator as the current segment, i.e., the same side of the contour generator plane as the current segment. In this example, segments (2, 3, 4, 5, 11, 12, 13, 14, 15, 16, 17, 25, 26) lie on one side of contour generator plane whereas (7, 8, 9, 10) all lie on the other side of the plane. When the current segment is 12 and the previous one is 11, the next segment should lie on the same side of the cg plane as 11 and 12, i.e., it should belong to the set (2, 3, . . . , 26) given above.

preferences

Having eliminated some segments using the above constraints, we may still be left with a number of candidates for the next segment. In this example we will be left with segments 13, 14 and 26. To choose the next segment, we use the following preference criteria (the weights used are given in parenthesis):

* Segments coplanar with the current and previous segments are preferred (weight=50000). In this case 13 and 14 will be preferred to 26, as they are coplanar with 11 and 12 but 26 is not.

* Segments lying on the circle in which previous segments lie are given a greater weight, as we prefer circular terminators (weight=25000). Again, 13 and 14 lie on a circle through 11 and 12 but 26 does not.

* A segment closer to the current segment is preferred to segments farther away (weight ranges from 0 to 1000). Here 13 is preferred the most and then 26 and 14.

* A segment more collinear with the current segment is preferred (weight ranges from 0 to 100). Again, 13 is more collinear with 12 than is 26 or 14 with 12 and is thus preferred over them.

Note that the first two criteria are assigned much more weight than the other two. However, the first two criteria are not mandatory as they may not hold and then we have to depend on the other two. The extent to which a segment is preferred is determined by adding the weights and comparing the net weight with the net weight for competing segments. The segment with the highest weight is selected as the next segment. In this example, both segments 13 and 14 satisfy the first two criteria but 13 gets a greater weight from the last two criteria and is thus selected as the next segment.

4.3. Choosing GCs

We now have several possible GC descriptions of the same volume and we need a method of rating them and choosing the better descriptions. For this we characterize the GCs by finding the attributes of the contour generator(s) and the terminator(s). The characteristics of cgs used are: their length and whether they are parallel or not. The characteristics of terminators used are: closed or open, planarity and circularity. If a planar terminator is found we compute its normals. If the terminator is circular, we compute the center and radius of the circle, else we compute its centroid and the average distance of the centroid from the boundary segments.

Preference measures are then computed for each GC based on its attributes. Associated with each attribute is a weight, indicating its relative importance. Note that these weights are order of magnitude numbers. We illustrate this method by considering the GC found in the example of figure 4 with cg1=1 and cg2=6 and terminators t1=(11, 12, 13, 14, 7, 8, 9, 10) and t2=(2, 3, 4, 5). Remarks for this example are made parenthetically.

1. Two parallel terminators -highly preferred-; weight=5000 (t1 and t2 satisfy this).

2. Planar, closed terminator; weight=1000 (t1 satisfies this)

3. Planar terminator, not closed; weight=500 (t2 satisfies this).

4. Closed terminator, not planar; weight=100 (this case doesn't hold here)

5. Circular terminator, in addition to being planar; weight=100 (both t1 and t2 satisfy this).

6. Terminators with more segments are preferred (they give more information about the cross-section); weight=10 per segment in terminator (t1 gets a weight of 10*8=80 and t2 gets a weight of 10*4=40).

7. Longer contour generators are assigned a higher weight than shorter ones (we prefer a GC description with a longer axis than one with a shorter one); weight=(length-of-cg)*100/(maximum-length-of- segments-in-the-scene) (assuming cg1 and cg2 to be the longest segments in the scene, we get a weight of 100 for each cg).

8. Parallel contour generators are preferred (we prefer cylinders over cones); weight=10 (cg1 and cg2 satisfy this).

9. Terminators normal to the contour generators are preferred; weight=100 (cg1 and cg2 are normal to t1 and t2).

The preference measure of each GC is found as the sum of the weights of its attributes. (In the GC of this example, the preference measure is 7020.) The GCs are then sorted according to their preference measures.

GCs that have the same segments are assumed to describe the same volume. Therefore, a GC that is disjoint from any GC of higher preference is treated as a separate object.

4.4. Describing The Objects As GCs

We would now like to describe the objects found in the scene in terms of GCs. This means that for each object we find the three GC functions, which are: the axis curve, the cross-section and the cross-section function. The axis curve is the 3-D locus of the centroid of the cross-sections of the GC. The cross-section function indicates the way the cross-section size changes as we go along the axis (Shafer's transformation rule). The cross-section indicates the cross-section shape. (Since we model objects as Right GCs, the cross-section plane is assumed to be normal to the axis.)

These functions must be deduced from the contour generators and terminators. We find the axis curve as the line of symmetry between the the contour generators. The cross-section function is defined by the distance of either contour generator from the axis. The cross-section shape is decided on the basis of the shape of the preferred terminator, if one or more has been found. A circular terminator is preferred the most, next a closed terminator and then one with a larger number of segments.

In the case of a circular terminator, only a part of which is seen (due to occlusion, say), we take the cross-section to be the complete circle. In the case when only part of one contour generator is seen, we hypothesize that the object extends as far as the longer contour generator

and then compute the axis curve and the cross-section function.

5. AN EXAMPLE TO ILLUSTRATE THE METHOD

We now illustrate the working of our algorithm with the example of figure 4. In this case the cg directed method explores $\binom{n}{2}=\binom{26}{2}=325$ possibilities (here n is the number of segments in the scene). Of these only 32 pairs have coplanar cgs and 7 of these have corresponding terminators. The method decides there is one object, the GC with cgs (1,6) and terminators (11, 12, 13, 14, 7, 8, 9, 10) and (2, 3, 4, 5), as it is the best GC and all other GCs are its subsets. This GC is given a very high preference measure because it has planar, parallel terminators, one of which is closed, and it has long, parallel cgs. Other GCs are found but discarded because their preference measures are lower and they are subsets of the best GC. For example, a GC with (2, 7) as cgs and (1, 10, 2, 8, 7) & (3, 4, 5, 6) as terminators, is also found but has a lower preference measure because of its short axis and non-planar terminators. It is a subset of the best GC and is thus discarded.

The terminator directed method first looks for coplanar sets and finds 37 of them. Of these, it finds cgs for just two of them. For the sets (7, 8, 9, 10, 11, 12 13, 14) {planar and closed} and (2, 3, 4, 5) {planar only}, the cgs found are 1 and 6. The corresponding GC is rated high. Other terminators like (1, 2, 11) are also found but the corresponding GCs are not rated high because the terminators are not closed and corresponding cgs cannot be found (the hypothesis cannot be verified). Here again, other hypotheses are rejected because they are poorer and are subsets of the best hypothesis.

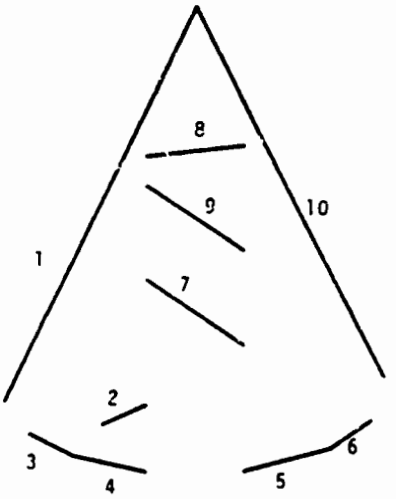
For this example, the two methods (cg directed and terminator directed) find the same hypothesis. The cg and terminator sets of this hypothesis are used to compute the GC functions, which are then used to generate slices for display.

6. RESULTS

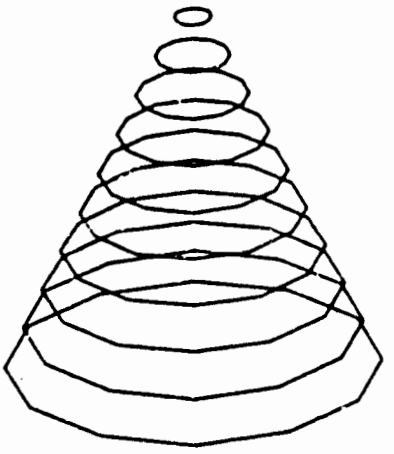
The above algorithm has been implemented as a computer program and has been tested on a number of synthetic images. We present a few of the results here.

Figure 5 (a) shows the sparse data for a cone with the segments numbered. It has a number of surface markings. The boundary is also sparse and there are segments missing at the terminator. The corners (junctions between terminator and cgs) are also missing. This is similar to the output of a stereo program. The cg directed method explores 45 possibilities, 24 of which have coplanar cgs and 3 of which have terminators. The terminator directed method finds just one coplanar set (with 3 or more segments) and the corresponding cgs for it. The best GC found has (1, 10) as the cg pair and (3, 4, 5, 6) as the corresponding terminator. The GC generated from the functions corresponding to this figure is shown figure 5 (b). We may note that although we see just a part of the cross-section, our program hypothesizes a complete circular cross-section and outputs a GC accordingly.

Figure 6 (a) shows the sparse data for a more complex scene, with several objects occluding one another, and with surface markings and missing segments. There are 64 segments here. The cg directed method explores 2016 possibilities of which about 445 have coplanar cgs and 35 of them have corresponding terminators. The terminator directed method finds 40 coplanar sets of which only three have corresponding cgs. The GCs found are shown in figure 6 (b).



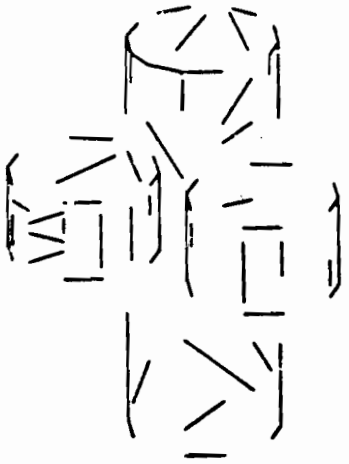
(a)



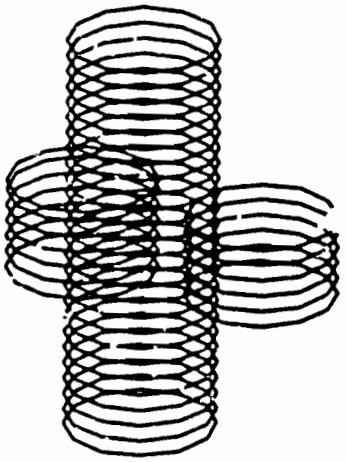
(b)

Figure 5: A Cone

(a) Sparse 3-D Data, (b) Corresponding GC



(a)



(b)

Figure 6: Several Objects Occluding One Another

(a) Sparse 3-D Data, (b) Corresponding GCs

7. CONCLUSIONS AND FUTURE WORK

We have presented an approach for segmentation and description of 3-D objects as generalized cones from sparse 3-D data in the presence of surface markings, noise and missing segments. Our algorithm has been tested so far with images of Linear Straight Homogeneous Generalized Cones with occlusion.

Our work represents only an initial effort in this direction and many important problems remain to be solved; some are given below:

- Our program does not handle non-linear non-straight homogeneous GCs. However, we believe we can extend the methods of LSHGCs by considering the non-linear non-straight homogeneous GCs as piecewise LSHGCs.

- Use of surface markings to strengthen the hypothesis of a GC.

- We have not explored control issues yet. How do we weigh the hypotheses of different methods, which to explore first and which to prune?

- The technique of finding a continuous curve amidst noise needs to be made more powerful.

- The data for horizontal/near horizontal segments is either missing or extremely sparse in the output of stereo programs. At these places we need to use the 2-D data, in addition to the available 3-D data.

References

1. Shafer, S.A., "Shadow Geometry and Occluding Contours of Generalized Cylinders," Tech. report, CMU Report CS-83-131, May 1983.

2. Nevatia, R., *Machine Perception*, Prentice Hall, 1982.

3. Binford, T.O., "Visual Perception by Computer," *IEEE Conference on Systems and Controls*, December 1971, .

4. Nevatia, R. and Binford, T.O., "Description and Recognition of Complex-Curved Objects," *Artificial Intelligence*, Vol. 8, 1977, pp. 77-98.

5. Marr, D., "Analysis of Occluding Contour," *Proceedings of the Royal Society of London*, 1977, pp. 441-475.

6. Brooks, R.A., "Symbolic Reasoning among 3-D Models and 2-D Images," Tech. report Memo AIM-343, Stanford Artificial Intelligence Laboratory, June 1981.

7. Grimson, W. and Marr, D., "A Computer Implementation of a Theory of Human Stereo Vision," *Proceedings of DARPA Image Understanding Workshop*, Palo Alto, Calif., April 1979, pp. 41-47.

8. Terzopoulos, D., *Multiresolution Computation of Visible-Surface Representations*, PhD dissertation, Massachusetts Institute of Technology, Departments of Computer Science and Electrical Engineering, January 1984.

9. Medioni, G. and Nevatia, R., "Segment-based Stereo Matching," *Proceedings of DARPA Image Understanding Workshop*, Washington, D.C., June 1983.

GC970145-53

EQUIVALENT DESCRIPTIONS OF GENERALIZED CYLINDERS

Kenneth S. Roberts*

* Department of Computer Science, Columbia University, New York, N.Y.

Abstract

The "equivalence" problem for shape descriptions is that a single three-dimensional shape may have several different descriptions. The Slant Theorem (Shafer[1]) for equivalent generalized cylinder descriptions was proven under the restrictions that the same radius function and the same axis be used for all the descriptions. A proof is given that the theorem still holds when the "same radius function" condition is removed. It does not hold when the "same axis" condition is removed. The ellipsoid is a counter-example.

Introduction

The equivalence problem for shape descriptions is that a single three-dimensional shape may have several different, equivalent descriptions. One way to deal with this problem is to use a method of generating descriptions which guarantees that the description produced is always a unique, canonical representation. The other approach is to permit alternate descriptions, but be able to tell when two descriptions are equivalent, i.e. describe the same shape

Shafer[1] investigated this second approach for a class of generalized cylinders. After eliminating the trivial equivalences due to rotation, etc., Shafer gave theorems about some families of equivalent descriptions.

The Slant Theorem

Following Shafer[1], a generalized cylinder is Straight if its axis (or spine) is a straight line segment. It is Homogeneous if all its cross-sections have the same shape except for scale. A Straight Homogeneous Generalized Cylinder (SHGC) is given by the four-tuple (A,C,r,α) (see Figure 1).

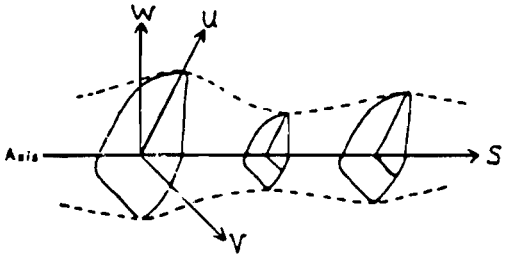


Figure 1: Straight Homogeneous Generalized Cylinder

A is a line segment in 3-space called the Axis or spine. It is parameterized in s, and an s-coordinate may be defined coinciding with the Axis. α is the (constant) angle of each cross-section plane to the Axis. C is the planar cross-section curve. Coordinates u and v may be defined for the cross-section plane, such that the u-coordinate coincides with the projection of the Axis onto the cross-section plane. C may then be parameterized in t: $C(t) = (u(t),v(t))$. r(s) is the radius function, which gives the scale of the cross-section at each point along the Axis (C gives the shape of the cross-section, r gives the scale). So in suv-space, each point on the surface is given in terms of parameters s and t by $(s, r(s)u(t), r(s)v(t))$. A mutually orthogonal set may be formed by replacing the u-coordinate with a w-coordinate perpendicular to the Axis and the v-coordinate. Then in swv-space, the point on the surface given by parameters s and t is $(s + r(s)u(t)\cos\alpha, r(s)u(t)\sin\alpha, r(s)v(t))$. In this paper, it is assumed that the Axis function A(s) is linear, and that the radius function r(s) and cross-section function C(t) are piecewise C^2.

An SHGC is a Right SHGC if its cross-section angle $\alpha = \pi/2$. Otherwise it is an Oblique SHGC. An SHGC is Linear if its radius function r(s) is linear. The Slant Theorem (Shafer[1], page 103 and Appendix E) states that:

> An Oblique Straight Homogeneous Generalized Cylinder (SHGC) has an equivalent Right SHGC if and only if the radius function of the Oblique SHGC is Linear. (Two otherwise equivalent descriptions which have differently sloped ends are regarded as equivalent for the purposes of this theorem).

The theorem was proven under the restricted conditions that the same radius function and same Axis be used for both the Oblique and the Right SHGCs. The question arises whether the theorem still holds when these conditions are relaxed.

The "same radius function" condition

The Slant Theorem still holds when the "same radius function" condition is removed. The "if" part of the theorem ("Linear radius function implies the existence of an equivalent Right SHGC") is already true from the restricted form of the theorem. So what must be proven is the following:

> Given an Oblique SHGC $G = (A,C,r,\alpha)$ where radius function r is non-linear there does not exist any Right SHGC $G^* = (A,C^*,r^*,\pi/2)$ which has the same Axis as G (without restriction on the radius function r^* of G^*).

Proof: The basic idea is that at least one of the angled cross-sections of the Oblique SHGC will be on a non-linear bend in the radius function r(s). But the bend must be spread over a wider range of cross-sections in the Right SHGC, and there is no way for

one radius function to consistently handle all of them.

Given an Oblique SHGC G = {A,C,r,α}, the "zigzag" construction shall be defined as follows for a value of s = s_2 and values of t = t_1 and t = t_M (see Figure 2). Call the point given by s = s_2 and t = t_L, L_2; similarly for M_2. Working in swv-space, the points may be displayed in a plot of w against s (see Figure 2). The coordinates of L_2 in swv-space are $(s_2 + r(s_2)u(t_L)\cos\alpha,\ r(s_2)u(t_L)\sin\alpha,\ r(s_2)v(t_L))$.

The set of points $(s + r(s)u(t_L)\cos\alpha,\ r(s)u(t_L)\sin\alpha,\ r(s)v(t_L))$ for all s, forms a curve in swv-space, call it "curve L" (likewise "curve M"). Take the plane in swv-space perpendicular to the Axis which

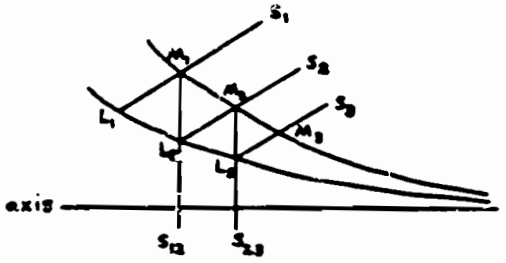


Figure 2: The zigzag construction

contains L_2. Call the intersection of that plane with curve M, point M_1. Call the s-value for that point s_1. Now take the plane which is at an angle α to the Axis and contains M_1. Call the intersection of that plane with curve L, point L_1. Similarly, work in the other direction to define s_3, L_3, and M_3 (see Figure 2, which plots only the w and s coordinates).

(For some SHGCs and values of s and t, it may be that the intersection of the curve L and the plane in swv-space may include more than one point, or even a line (but not less than one point). In such cases, it is fairly easy to see that all the cross-sections perpendicular to the Axis cannot have the same shape, in which case no Right SHGC can be constructed which is homogeneous, and the theorem is satisfied. So in what follows it will be assumed that each point in the Oblique cross-section maps to exactly one point in the Right cross-section.)

Since r(s) is non-linear, there exists some value s = s_2, some ε > 0, and some real value m, such that for all s in $(s_2 - \varepsilon, s_2)$, the slope of r(s) is less (greater) than m, and for all s in $(s_2, s_2 + \varepsilon)$, the slope of r(s) is greater (less) than m.

It is evident that by choosing t_L and t_M close enough together, L_2 and M_2 can be chosen with $u(t_L)$ and $u(t_M)$ close enough together so that the zigzag construction can be made with s_1 in $(s_2 - \varepsilon, s_2)$, and s_3 in $(s_2, s_2 + \varepsilon)$. Further, this can be done so that $u(t_L)$ and $u(t_M)$ are equal neither to each other nor to 0, because SHGC G is a closed figure. (As an example of a non-closed figure which does not satisfy this theorem, take as the cross-section a line segment parallel to the v-coordinate, and any reasonable non-linear radius function).

From the way in which this zigzag construction has been done, it is clear that the slopes of line L_1L_2 and line L_2L_3 are not equal (likewise for line M_1M_2 and line M_2M_3).

Lemma: The slopes of line L_1L_2 and line L_2L_3 are equal (likewise for line M_1M_2 and line M_2M_3) if and only if $r(s_1)/r(s_2) = r(s_2)/r(s_3)$.

Proof: Using (s,w) coordinates (and ignoring the v-coordinate), it can be seen from the way in which the zigzag construction is done that

$$L_1 = (s_1 + r(s_1)u(t_L)\cos\alpha,\ r(s_1)u(t_L)\sin\alpha).$$

$$M_1 = (s_1 + r(s_1)u(t_M)\cos\alpha,\ r(s_1)u(t_M)\sin\alpha)$$

$$L_2 = (s_2 + r(s_2)u(t_L)\cos\alpha,\ r(s_2)u(t_L)\sin\alpha).$$

But also

$$L_2 = (s_1 + r(s_1)u(t_M)\cos\alpha,\ r(s_2)v(t_L)\sin\alpha).$$

So the slope of line L_1L_2

$$= |r(s_1)u(t_L)\sin\alpha - r(s_2)u(t_L)\sin\alpha| \ / \ |(s_1 + r(s_1)u(t_L)\cos\alpha) - (s_1 + r(s_1)u(t_M)\cos\alpha)|$$

$$= \tan\alpha\ |u(t_L)/(u(t_L)-u(t_M))|\ |1 - (r(s_2)/r(s_1))|$$

Likewise it can be shown that the slope of line L_2L_3

$$= \tan\alpha\ |u(t_L)/(u(t_L)-u(t_M))|\ |1 - (r(s_3)/r(s_2))|$$

And the result follows (with the same argument for M_1M_2 and M_2M_3).

Now using the Lemma, we get the result that $r(s_1)/r(s_2) = r(s_2)/r(s_3)$.

But if G° were a valid SHGC, with its radius and cross-section functions r°(s) and C°(t), the following would hold:

$$r(s_1)u(t_M)\sin\alpha \ / \ r(s_2)v(t_L)\sin\alpha$$

$$= r^\circ(s_{12})u^\circ(t_M^\circ) \ / \ r^\circ(s_{12})u^\circ(t_L^\circ)$$

$$= r^\circ(s_{23})u^\circ(t_M^\circ) \ / \ r^\circ(s_{23})u^\circ(t_L^\circ)$$

$$= r(s_3)u(t_M)\sin\alpha \ / \ r(s_3)u(t_L)\sin\alpha$$

where the middle equality is due to the "Homogeneous" part of "SHGC" (all cross-sections must have the same shape, up to scale). These equalities would imply that $r(s_1)/r(s_2) = r(s_2)/r(s_3)$. Since this has been shown to be false, no equivalent Right SHGC can exist.

The "same axis" condition

The theorem does not hold if the "same Axis" restriction is removed. If different axes are permitted, then there are non-Linear SHGCs that have different, equivalent SHGC descriptions. The sphere is a trivial counter-example that will not be considered, since its alternate descriptions differ only by rotation.

But there are non-trivial counter-examples: Consider a right ellipsoid with center at the origin in Cartesian 3-space. It can be represented in equation form as:

$$x^2/a^2 + y^2/b^2 + z^2/c^2 = 1$$

Thinking in terms of generalized cylinders, and taking the x-axis as the Axis, we have a Right Non-linear SHGC, with elliptical cross-sections.

Now suppose we slant the Axis by an angle α in the x-y plane, but leave the elliptical cross-sections parallel to the y-z plane (a kind of skew transformation). This "oblique" figure is clearly an Oblique Non-linear SHGC, again with elliptical cross-sections. This "slant" transformation can be carried out in the equation representation by replacing y with $y - x \tan \alpha$ and rearranging to get:

$$x^2(1/a^2 + \tan^2\alpha / b^2) - xy(2 \tan \alpha / b^2) + y^2/b^2 = 1 - z^2/c^2$$

Analytic geometry texts show that the left side is the equation of a family of ellipses that have been rotated in the x-y plane by an angle

$$\beta = (1/2) \arctan[2 \tan \alpha / (1 - b^2/a^2 - \tan^2\alpha)]$$

These ellipses are centered on the z-axis, and it is easy to show that their orientation and eccentricity is independent of the value of z. They all have the same shape. So this "oblique" figure may be represented as a Right Non-Linear SHGC, with Axis on the z-axis, and elliptical cross-sections.

This type of result is not limited to ellipsoids. But the ellipsoid has this additional property: the "oblique" figure is simply another right ellipsoid, rotated from the x-axis by the angle β given above. If the rotation by β is carried out on the equation representation, the result is:

$$(x^2/a^2)[\cos^2\beta + (a^2/b^2)(\tan^2\alpha\cos^2\beta - 2\tan\alpha\cos\beta\sin\beta + \sin^2\beta)]$$

$$+ (y^2/b^2)[\cos^2\beta + 2\tan\alpha\cos\beta\sin\beta + \tan^2\alpha\sin^2\beta + (b^2/a^2)\sin^2\beta]$$

$$+ (z^2/c^2)$$

$$= 1$$

The eccentricity is different from that of the original right ellipsoid, as we would expect.

So the "oblique" figure can also be represented as a Right non-Linear SHGC with the Axis in the x-y plane at angle β. Thus "being a right ellipsoid" is a non-Linear property of SHGCs which is invariant under skew transformations. To put it another way, there is no such thing as an oblique ellipsoid.

It is interesting that while the z-axis representation depends on being able to take advantage of the freedom to orient the Axis anywhere in three dimensions, the "angle β" representation also works as a counter-example to the two-dimensional analog of the Slant Theorem.

So there are some Non-linear Oblique SHGCs which are equivalent to Right SHGCs, and therefore the "only if" part of the Slant Theorem does not hold without the "same axis" condition.

Families of descriptions with different axes

Define an H-axis for a shape as a line which is the Axis for some SHGC description for that shape. An RH-axis is an Axis for some Right SHGC description. Shafer in his Pivot Theorem ([1], p. 105 and Appendix F) has described families of H-axes which all use the same cross-sections, which exist only for Linear SHGCs. Other classes of shapes that have multiple H-axes:

1. There is an H-axis lying in the x-y plane, and the equation representation for the shape can be written in the form

$$f(x,y) = g(z)$$

and f satisfies

$f(kx,ky) = h(k)f(x,y)$ for some function h.

Then the z-axis is an RH-axis. For example:

$$(x/a)^4 + (y/b)^4 + (z/c)^4 = 1$$

2. The cross-section of a Right SHGC itself has multiple H-axes in its plane. For example, a square has four H-axes, a regular pentagon five, and a circle infinity. Any radius function can be used. These also satisfy the property of the previous type above. Included here could be spheres, cylinders, right prisms with polygonal bases, tetrahedra, octahedra.

3. Various elongations and skews of the first two types. Included here would be oblique prisms with polygonal bases and irregular tetrahedrons. The ellipsoid can be seen as an elongated sphere.

REFERENCES

[1] S. A. Shafer, *Shadow Geometry and Occluding Contours of Generalized Cylinders*. PhD dissertation, Carnegie-Mellon University, May 1983.

This research was supported by DARPA contract N00039-84-C-0165 and by an AT&T Bell Laboratories graduate scholarship.

GC970145-54

The Calibrated Imaging Lab Under Construction at CMU

Steven A. Shafer
Computer Science Department
Carnegie-Mellon University
Pittsburgh, PA 15213
5 November 1985

Abstract

This document describes the Calibrated Imaging Laboratory, a facility for precision digital imaging under construction at CMU. The purpose of this lab is to provide images with accurate knowledge about ground truth (concerning the scene, illumination, and camera) so that computer vision theories and methods can be tested on real images and evaluated to determine how accurate they really are. The lab aims to provide ground truth data accurate, in the best circumstances, to the nearest pixel geometrically and the nearest 8-bit pixel value photometrically. There are also many illumination and imaging facilities in the lab that provide increased flexibility or increased complexity of the visual situation, at a cost of reduced precision in the ground truth data.

To accomplish these goals, the lab includes mechanisms to carefully control and measure the direct and indirect illumination in the scene, the positions of objects, and the properties of the camera/digitizer system. Lighting can be provided by a near-point source (5 mm diameter aperture) for high precision, or by a general-purpose track lighting system for flexibility. The work area can be surrounded by black curtains etc. to reduce stray light and indirect illumination. The cameras include a very high-precision CCD camera on a static mount, and an X-Y-Z-pan-tilt jig with multiple inexpensive CCDs aligned with each other. Surveyors' transits are used to measure positions of points in space, and other calibration materials are available for all types of camera property measurement. Color imaging by serial selection of filters is also available.

The lab is described as we currently envision it will be equipped when the facilities are operational; the current status is summarized at the end of the paper.

1. Introduction

Computer vision research encompasses both theoretical and empirical studies, but there has been a mismatch between the two. Current theories typically depend upon unrealistic assumptions, making it impossible to test them on real images. Because they are only tested on synthetic data, most theories cannot be analyzed to determine exactly what the sources of inaccuracy and error might be in analyzing real images. Likewise, empirical studies using real images usually cannot apply existing theories, so they rely on guesswork and heuristics instead.

We are building a Calibrated Imaging Lab (CIL) at CMU to address both aspects of computer vision by bridging the gap between theories and real images. The CIL is a facility for capturing images with high precision in a controlled environment. It will make possible the study of both geometric and photometric computer vision theories with high-fidelity images and good knowledge of ground truth.

High-precision images are important in computer vision studies because they make it possible to evaluate the results of numeric algorithms such as stereo or motion analysis. With precisely known imaging geometry and object locations, it is possible to look at the output from an algorithm and tell how accurately it performed rather than relying on a coarse, subjective evaluation. It is also possible to pinpoint sources of systematic error or bias that are too subtle to be identified when the ground truth or imaging geometry are unknown or imprecise.

Control of the imaging environment is also important, especially for photometric analysis of images. Photometric methods are notorious for depending upon assumptions about the lighting, background illumination and reflections, and camera pixel values that are rarely found in practical situaions. In fact the assumptions usually made in such research are extremely difficult to match in practice, which may help to explain why so few photometric methods have been successfully applied to real images. In order to evaluate current theories and develop new theories based on more realistic assumptions, it is necessary to have control over the direct and indirect illumination in the image and to know the precise photometric response of the camera.

In the Calibrated Imaging Laboratory (CIL) at CMU we are addressing both the need for precise images and the need for a controlled environment. For precise imaging, the CIL has the goal of providing images accurate geometrically to the nearest pixel location and photometrically to the nearest 8-bit pixel value. For controlling the environment, the CIL has the goal of providing a continuum of imaging situations ranging from a (near-)point light source with no background illumination and a static high-precision camera, to commercial lighting fixtures with bright walls and a movable platform for several low-cost cameras. A range of complexity can thus be achieved in the CIL along several different dimensions of the imaging situation.

It is our intention to use the CIL for several purposes:

- For testing existing theories for image understanding in these areas:
  - Shadow geometry and other geometric shape inference methods
  - Stereo (using two or more cameras), motion, and combined motion stereo analysis
  - Photometric and reflectance map methods for shape recovery
  - Color analysis of gloss and surface material

- For producing new theories dealing with more complexity (and hence more realism) in these areas:
  - Camera distortions
  - Non-uniform illumination and extended light sources
  - Substantial background (diffuse) illumination
  - Glossy and other non-Lambertian surfaces
- For evaluating the properties of commonly available cameras used for machine vision.
- For providing access to high-precision image data to interested researchers at CMU and elsewhere.

1.1 Overview

This informal paper describes the facilities of the CIL. These facilities include:

- *Lighting Control* provided by a near-point light source (arc lamp) for precision shadow analysis, and a complete track lighting system for flexible general illumination.
- *Background Reflection Control* in a room with black ceiling, black carpet, and black or white curtains, with other colored backdrops as needed.
- *High-Precision Color Images* provided by a custom-built camera yielding 512x512x8 images with each pixel value being repeatable (noise-free) and linearly related to scene radiance, using color filters in a filter wheel.
- *Precision Stereo and Motion Image Sets* provided by a mobile platform with precision X-Y-Z-pan-tilt controls and a pair of CCD cameras aligned for stereo correspondence.
- *Objects* including simple objects for viewing and a scale model landscape that presents a variety of surface property, motion, and occlusion situations.
- *Calibration Data* provided by appropriate tools, including photometers, precision targets, and calibration camera filters.
- *Accurate Ground Truth Data* provided by an optical table with precision position control devices and surveyors' transits for position measurement.

Although the paper is written in the present tense, the CIL is not yet in complete existence. A status report on the current state of the lab is included in this paper.

2. Lighting Control

Figure 2-1 shows the overall layout of the CIL. The "business area" is the optical table, 4 x 8 feet, with cameras at one end looking at the "scene" at the other end. Lighting is provided by a track lighting system overhead or an arc lamp on a stand; the whole work area is enclosed in curtains to eliminate outside light. A desk and storage area complete the lab (which is already straining the size of the room!).

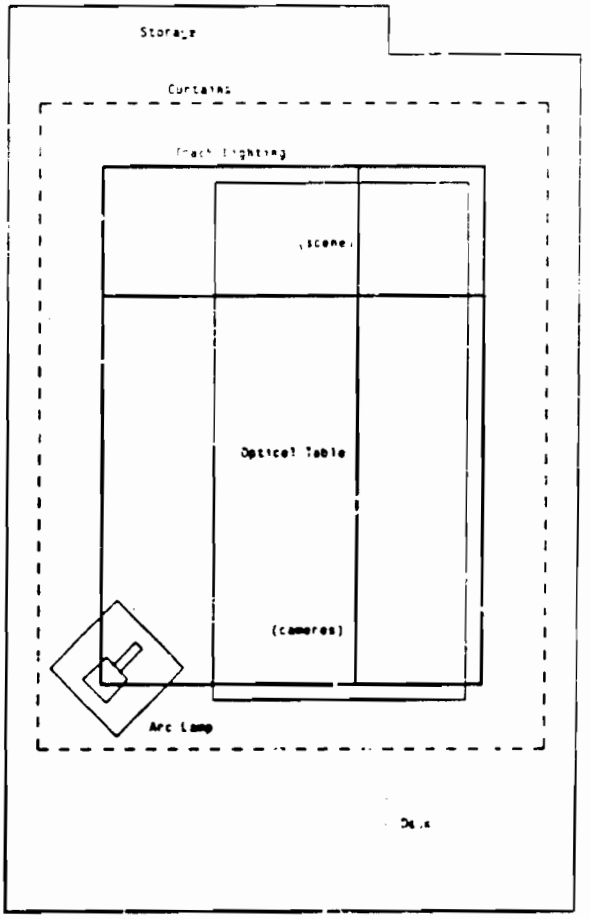


Figure 2-1: Layout of the Calibrated Imaging Lab

2.1 Direct Lighting by Normal Fixtures

Ordinary illumination for imaging is provided by a ***track lighting system*** in the ceiling. The track layout allows many configurations of light sources which may be directly over the scene or camera, or at virtually any angle and direction. Vertical positioning is provided by wands that attach to the track and hang downward, allowing fixtures to be mounted at approx. 1-foot intervals from the ceiling height down to about 4 feet off the floor. Each fixture itself has a gimbal allowing imprecise pan and tilt control (except the fixtures for the spherical "globe" bulbs which simply point downward).

We have a variety of fixtures, bulbs, and attachments. Of primary interest are spot and flood lamps of various wattages up to 500 W, mountable on gimbal fixtures with cowlings or reflectors. We also have a variety of grilles and (uncalibrated) color filters for lighting experimentation, and other bulbs such as globes.

The track itself has four circuits, each with a separate switch on the wall. When a fixture is attached to the track, an adaptor is used whose position and switch setting determine the circuit that will power the fixture. We use one circuit for normal room lighting, two for alternate imaging illumination configurations, and one for auxiliary power when multiple high-wattage bulbs are to be used.

We also plan to obtain a couple of vertical rectangular ***light panels*** for studying light sources of various shapes, and we can easily obtain ***fluorescent tubes*** to hang from the track.

2.2 Direct Lighting by Point Source

High-precision illumination control is provided by a "point source", an ***arc lamp*** with a very small aperture. This lamp was configured with the following specifications assuming the scene is 1 m deep and is 3 m from the source:

- Shadow edges on surfaces facing the camera should be no more than one pixel (about 1 mm) wide.
- The center of the illuminated spot should receive 1000 lux (100 foot candles) of light, equivalent to recommended office illumination for reading.
- The spot should be at least 4 feet wide, with at least 500 lux at the edge of the spot. (Uniform illumination would be nice but is too demanding a specification.)

These specifications give rise to the following requirements:

- The aperture must be less than 1 cm in diameter.
- The lamp must be a 1000 W arc lamp.
- The condensing lens must have *f*-number no greater than 2.

We have such an arc lamp system, as configured (optically and mechanically) by the manufacturer. It consists of a lamp housing 18 inches high, with a lens and precision 5 mm aperture in front; these pieces are all mounted on a base plate 9 x 20 in. The assembly weighs about 20 lb. and has a power supply attached by several cables.

To support the arc lamp, we have a scaffold 6 feet high made of pipes, with a movable platform attached to the base plate of the arc lamp. The platform can slide vertically from 3 to 6 feet above the floor, and can be tilted within ±45° from horizontal. The scaffold can sit on a dolly to be pushed about on the floor.

Unfortunately, the arc lamp has some associated safety hazards: high pressure of the lamp, a high-voltage ignition pulse, ultraviolet light emitted by the lamp, and the high intensity of the source itself. We have precautions such as a UV filter for the lamp and goggles to wear, and a key switch for the power supply, but use of the arc lamp will be minimized in the lab because of the hazards.

2.3 Diffuse Illumination Control

For control of the background illumination, we would like a totally non-reflective surrounding. In practice, we must accept a rather poor solution due to the cost and maintenance problem of such an environment. We settle for black, relatively matte ***curtains***, a similarly painted ***ceiling***, and a dark gray ***carpet***. We also have white, fairly shiny curtains as an alternative for studying high-reflectance backgrounds. We can always use ***backdrops*** or hang other fabrics to achieve other effects.

The working area itself consists primarily of components painted black, but since the optical table top is gray metal, we may need to use black fabric pieces to minimize unwanted reflections from the table top

3. Imaging Facilities

Figure 3-1 shows the layout of the working area, the optical table containing the cameras and "scene" (the objects being viewed). The ***optical table*** itself is a 4 x 8 foot metal table that is extremely rigid, with a precise grid of 1/4-20 threaded holes on 1-inch centers across the entire surface. Cameras are mounted on two platforms

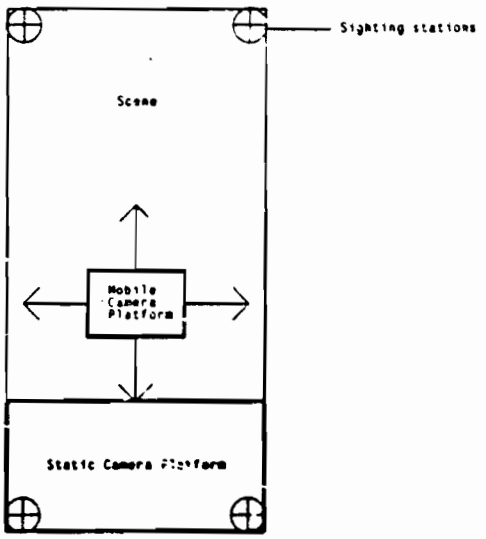


Figure 3-1: Layout of the Optical Table

on the table: the Mobile Camera Platform (MCP) for high-precision camera motion studies and the Static Camera Platform (SCP) for high-precision photometric studies.

3.1 The Mobile Camera Platform

The ***Mobile Camera Platform*** (MCP) is shown in figure 3-2. It provides high-precision X, Y, Z, pan, and tilt motion for a platform 12 x 12 inches square, with the rotation (pan/tilt) center placed at a point 7 inches above the center of the platform. This is accomplished by three components: an X-Y-Z jig, a pan/tilt assembly, and position control for each camera mount placed on the platform.

The X-Y-Z jig is constructed from long-travel translation stages with crank controls and mechanical (odometer-style) position readouts to 1/1000 inch resolution. Since a pixel (at 1 m with a normal lens) is roughly 1/50 inch across, this is more than adequate accuracy. The assembly has a 36 x 36 inch base providing 24 x 24 inch travel horizontally, with a vertical "picture-frame" assembly 32 inches (high) x 36 (wide) by 6 (thick) on the front edge of the base that moves vertically through 12 inches of travel.

Pan and tilt are controlled by high-precision manual rotation stages with direct measurement to .02° (about 1 arc-minute) of resolution and accuracy, with the ability to point in any direction (360° pan and 180° tilt).

Each camera has a mount with X-Y-Z-pan adjustment on top of the camera platform. This allows fine alignment of the cameras so the centers of perspective expansion can be placed in the plane of the pan and tilt axes and the optical axes of the cameras can be placed normal to this plane. With appropriate alignment, epipolar lines can be made to be scan lines. X-Y-Z translation will correspond to image and depth translation, and pan/tilt rotation will not induce any translational components. Roll or tilt adjustments can be inserted into each camera mount if needed. Alignment of the various positioning stages will be a complex task, but we have the necessary degrees of freedom and viewing targets if only we supply the patience.

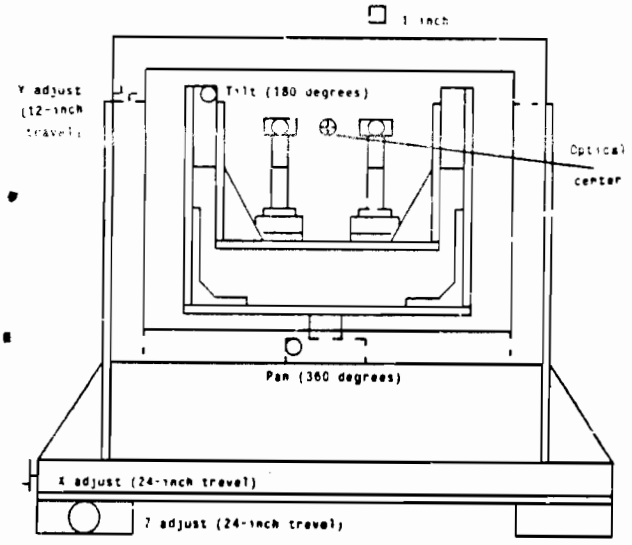


Figure 3-2: The Mobile Camera Platform

We have, however, noted a problem with alignment of cameras as described above: when the focus of a typical camera lens is adjusted, the center of perspective expansion moves along the optical axis. The cameras must then be re-positioned along the viewing direction if the perspective centers are to be kept in the plane of the pan and tilt axes. This problem does not exist at all for, say, outdoor vision with wide-angle lenses fixed at infinite focus, it only arises where close-up photography imposes depth-of-field restrictions that require changing the focus of the lenses from time to time.

There will be anywhere from ***one to four cameras*** mounted at any time on the MCP. Normally, we expect to use a pair of very inexpensive CCD cameras, with local modifications to the electronics that provide about 6 bits per pixel of photometric repeatability with excellent geometric fidelity. We will measure the photometric linearity ourselves if needed. Video facilities are shown in figure 3-3; they include ***one monitor for each camera***, a manual ***crosspoint video switch*** with inputs from the cameras on both the Mobile and Static Camera Platforms and outputs to a video patch panel for digitizing on any of our VAXes, and to a SUN with one or more digitizing frame buffer cards. We have a variety of ***lenses*** on hand.

The current MCP configuration has manual translation and rotation stages on the jig and a manual video crosspoint for digitizing; we have considered motorizing the entire facility. If we undertake real-time tasks in the lab in the future, there might be justification for such a considerable expense. By replacing manual with motorized components, we could obtain translation to .001 inch resolution at 3 in/sec, and rotation to .01° resolution at 30 deg/sec. Digitization might be wired up with parallel lines for real-time stereo.

3.2 The Static Camera Platform

In addition to the Mobile Camera Platform, there is a ***Static Camera Platform*** (SCP) intended for high-precision photometric control. The SCP itself is simply a 2 x 4 foot platform elevated 20 inches above the optical table, occupying one end of the table. The 2-foot depth will allow virtually any combination of color filters, polarizers, etc. to be placed in front of the cameras on the SCP.

The featured camera on the SCP is a ***high-precision camera*** custom-built for us to the following specifications:

- The output image is 512 x 512 x 8 bits, with square pixels.
- Repeated images of the same scene should produce the same pixel value at every location in the image (i.e. all 8 bits should be noise-free).
- Pixel values should be linearly related to radiance in the scene, and an image of a uniformly radiant scene should produce the same pixel value at every location (except a small number of known blemishes).

We do not require that the image be blemish-free, nor that it be geometrically accurate. However, the blemishes must be few in number and at measurable pixel locations, and the geometric properties must be stable over time. We will also allow the "image" to be the result of some constant-time processing upon the camera output, in particular for photometric linearity calculations which are based on constants that are unique for each pixel location.

The camera we have is based on technology used for astronomical cameras. Our supplier, an engineering firm, built a camera which differs from standard CCD CCTV cameras in several ways:

- The sensor chip is very high quality.
- The sensor chip is in an electric cooler just below 0°F, which reduces thermal noise to the required level for 10-bit noise-free digitization.
- The scanning rate of the CCD is slow (5 frames/sec, non-interlaced) to keep a high signal-to-noise ratio.

A digitizer of appropriate quality is attached, and the output is fed into an ***IBM PC*** by DMA to the PC memory. The 10-bit image data is then transformed into an 8-bit image by a photometric correction calculation, transforming each pixel value by a linear scaling with the coefficients for each pixel stored on disk on the PC. The engineering firm provided the photometric calibration data on a PC disk based on measurements made at its shop. The camera head is about a six-inch cube, with a standard C-mount for lenses; there is a separate power supply, and the digitizer occupies two cards in the PC.

We have a full set of ***color filters*** for both color separation (R-G-B) and bandpass filter work. These filters can be individually mounted in front of the lens, or mounted on a 4-position ***filter wheel*** used for the high-precision camera. We have ***IR filters*** on hand for use with uncorrected CCDs although many CCDs today for closed-circuit TV use have built-in filters to correct the spectral responsivity to be close to the human spectral luminous efficiency curve $V(\lambda)$. These filters are available for use on the Mobile Camera Platform as well, although space limitations on the MCP probably make it mandatory to use individual filter holders instead of filter wheels.

We can mount other cameras on the SCP at will. Using the high-precision camera as a standard of measure, the performance parameters of other cameras can be determined in highly controlled situations. This may give us data for field use of the same (less expensive) cameras, and will allow us to make more informed camera purchasing decisions in the future.

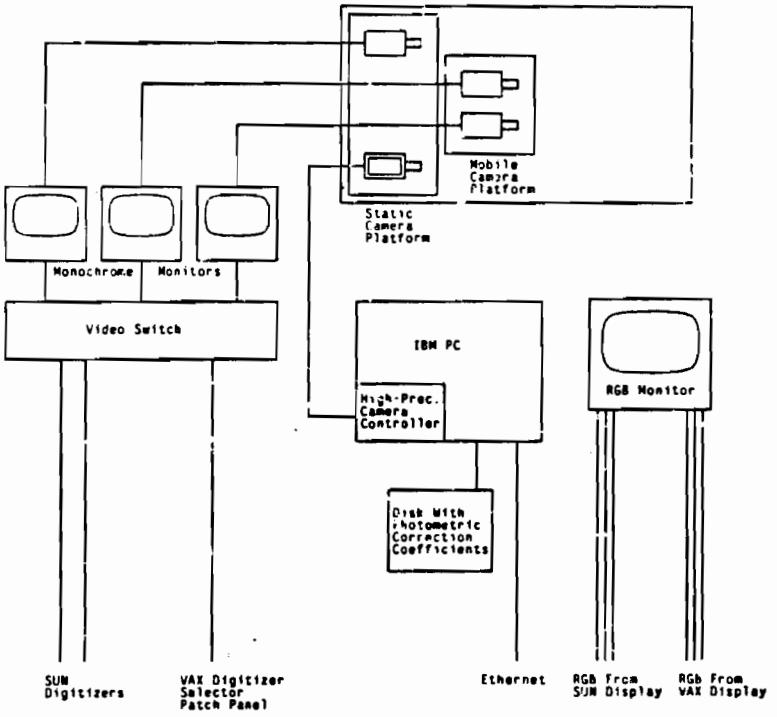


Figure 3-3: Video Facilities in the CIL

4. Other Facilities Related to Imaging

In the lab are a color monitor and a terminal, a monochrome monitor for each camera, the video crosspoint described above and shown in figure 3-3, and outgoing wires to the VAX computer network in the CMU Computer Science Department and the SUN and IBM PC used for digitization.

On the VAXes and SUNs we have a full library of ***pixel access and image manipulation*** software written in C; these machines run the UNIX operating system. LISP interfaces will be written under the auspices of the ALV project.

We are writing a program called the ***Geometric Calculator*** that acts like a combination desk calculator and database for manipulating geometric entities such as points, lines, vectors, planes, rotations, and coordinate frames. The basic operation in this program is to type in an expression and the program will calculate the result and print it out or store it with a name for later use. For example, an expression might be "the plane passing through three points: the point named X; the point defined by coordinates (R,S,T) in coordinate system C; and the point formed by the intersection of the line named L with the line through named point P in a direction parallel to the X-axis of coordinate system D". (Of course, the notation will be much more arcane than this!) The Geometric Calculator program, given this expression, would evaluate it and either print the result or store it with a name given by the user; collections of named objects can be stored in files and later recalled for further calculation. The program can deal with Cartesian as well as spherical coordinates, which are needed for the mensuration facilities described below. We may add mechanisms for perspective projections, possibly including some distortion, so that pixel location coordinates can be used directly as operands to the Geometric Calculator program. Objects are represented relative to coordinate systems which may move relative to each other; in this way, when a coordinate system is changed, all the objects and coordinate systems defined relative to it will also change (in relation to world coordinates) automatically.

In addition to simple objects to view, presenting a variety of textures, colors, and glossiness, we have a ***scale model landscape*** built with N guage (1/160) electric train models on a 3 x 5 foot board. The landscape is fairly complex, with a variety of simulated ground covers, terrain features, railroad track features, and buildings. The central area of the layout allows interchangeable plates to be inserted containing different types of buildings such as urban buildings, a rural scene, an oil refinery, or an industrial setting. The layout, building plates, and rolling stock have been carefully selected to yield a very wide variety of:

- *surface textures* -- smooth and shiny, smooth and matte, grainy, or with resolvable texture patterns
- *shapes* -- domes, generalized cylinders, blocks, sloping surfaces, protrusions of various types, symmetries and skewed symmetries
- *motion* -- motion of the train (usually moved by hand a precise distance between frames), of fixtures on the train (such as a crane), of vehicles or fixtures in the background, of several objects at once
- *occlusion* -- train crossovers, parallel tracks and coplanar tracks at angles to each other, tracks running through tunnels or bridges with frameworks, vehicles passing behind buildings or poles

This layout provides a very rich visual environment for all kinds of geometric imaging studies. There is also a great deal of tiny detail so that, for example, an interest-point operator looking at an image

of a boxcar will probably find dozens of feature points rather than the four or six points that would be found in a simpler domain such as the blocks world. Unfortunately, because of the scale, almost all flat surfaces will be made of plastic and thus unsuitable for serious studies of material type identification.

5. Mensuration and Calibration

The CIL has stringent mensuration (position measurement) and calibration requirements due to the stated goal of providing images accurate geometrically to the nearest pixel location and photometrically to the nearest bit of the pixel value.

5.1 Mensuration

The CIL must provide both coarse and fine measurement and control of the positions of objects, cameras, and lights in order to meet its geometric precision goals. The coarse position of objects is controlled by the optical table itself, which has a grid of 1/4-20 threaded holes on 1-inch centers across its entire surface. We have a collection of ***rods*** to use for supporting objects as needed, with accuracy provided only by rulers and levels. Fine position control can be obtained by ***optical-quality rotators and translators*** to move objects as desired. For the track lighting system, we have ruled scales mounted on the track itself, but such positioning is still rather coarse. The arc lamp on its dolly is also only positionable with ruler accuracy.

However, while we cannot obtain extremely fine control of the position of everything, we can and do *measure* positions to great accuracy. We do this using a set of ***electronic theodolites*** (surveyors' transits) mounted on the corners of the optical table; these are telescopes with crosshairs on pan/tilt mounts, knobs for fine aiming of the scope, and a digital readout of horizontal and vertical angle of the scope. The theodolites have an accuracy of 20 arc-seconds in both vertical and horizontal measurements, providing measurements accurate to approximately 1/3 pixel width at working distances. Our theodolites have machine-readable output jacks, although we have no current plan to interface them to our computers electronically. We have little stick-on bullseyes that can be used to create sighting points on a featureless surface.

It requires about 30 seconds for a person to obtain a measurement of a point on one theodolite, or about 2 minutes all told to measure a point in space and record its coordinates to the Geometric Calculator program. At this speed, it is not possible in general to exhaustively measure a complex object or illumination setting. It is our plan to use rulers to measure objects, then use the theodolites to determine exactly where those objects are placed on the table for viewing and to verify the most critical of the ruler measurements. Another problem with the theodolites is that their minimum focusing distance is about 50 inches, which means that we need four theodolites to cover the entire table top.

5.2 Calibration

The CIL will undertake several kinds of calibration to ensure the desired accuracy and precision goals. The most obvious calibration need is for determining the geometric projection properties of the cameras in use, which will be accomplished using a ***precision grid*** standing vertically up from the optical table. Our grid is 75 x 75 cm, with lines 1 mm thick forming 1 cm boxes. The precision of the line widths and spacings is 0.1 mm. We can place the grid on the table, measure a few points with the theodolites to determine the precise 3D coordinates of the grid vertices, and take a picture of the grid. A second picture with the grid at a different distance from the camera then provides sufficient information to calculate the geometric projection properties of the camera. Of course, such a projection is dependent on the lens and the distance at which the lens is focused, as discussed above. Thus, both positions of the grid must be within the depth of field of the camera without adjusting the focus. The precision grid will also be used for aligning cameras with each other and with the rotation axes on the Mobile Camera Platform.

For photometric calibration, we will place considerable reliance on the data supplied with our high-precision camera. Normally, we will use that camera as our working standard for evaluating other cameras; from time to time we can check on the calibration of that camera and have it recalibrated if need be. For routine photometric measurement, we have ***step-reflectance targets*** and ***neutral-density filters***. We also have a few ***color charts*** and a set of the ***British Ceramic Colour Standard tiles***, a set of 12 tiles 4 x 4 inches with a variety of colors and reflectances, whose spectral reflectance curves are known to within 1/2% at any wavelength. We have a hand held ***luminance meter*** (1/3° cone) and a hand-held ***incident colorimeter*** (hemispherical incidence, cosine-weighted) for routine use.

We also have on hand a 6-inch diameter ***Lambertian diffuser*** and a pair of 2 inch diameter ***polarizers***. The polarizers will be needed for studies of the effect of polarization on pixel values of various image sensors, we suspect that this effect is substantial near the edges of the image. The polarizers can also be used for studying polarized illumination by placing them in front of the arc lamp.

In an important sense, all geometric calculations in the CIL are only as accurate as the levels in the theodolites. For this reason, extra care was taken to ensure good leveling of the theodolites on the optical table and of the surface of the optical table itself.

6. Status and Future of the Lab

6.1 Applications of the Lab

We have many application areas in mind for the Calibrated Imaging Lab, both for implementing and evaluating current computer vision theories and for developing a new generation of more sophisticated theories. One such research area is the study of ***shadow geometry***. We plan to implement existing theories of shadow interpretation in a polyhedral environment and we are now working on the theory for interpreting "fuzzy" shadow edges caused by extended (non-point) light sources, using both geometric and possibly photometric methods. The CIL provides adequate facilities for generating all kinds of shadow problems in images, including very complex situations such as the shadows of moving objects.

Photometric studies and ***color analysis studies*** can also be carried out in the CIL. We would like to implement standard reflectance map methods as well as existing theories for color and gloss analysis. Then we can use these techniques to develop new ones that deal with more complexity and realism in images, such as non-trivial background illumination and non-uniform scene illumination. These studies may yield methods for identifying surface material type and roughness as well as surface shape information.

The Mobile Camera Platform in the CIL will be the basis for gathering high-precision images for binocular and multi-ocular ***stereo*** image sets as well as ***motion sequences*** and even ***motion stereo sequences***. We are developing analysis methods for motion sequences that we believe will yield substantial accuracy improvements over previously employed theories, and precise

ground truth data will be important in carrying out such a comparative evaluation.

Finally, the CIL will provide the necessary facilities for carefully measuring *camera properties*. This will allow a means for obtaining high-precision images, a means for measuring the accuracy of individual inexpensive cameras before using them in the field, and a base of general knowledge about cameras for machine vision that will allow more informed camera purchases in the future.

We hope that researchers outside of CMU will also benefit from the existence of the CIL, possibly by coming to obtain high-precision data (which will not be a trivial process, however).

6.2 Anticipated Difficulties in Using the Lab

For all its precision and flexibility, the Calibrated Imaging Lab presents some practical difficulties to deal with. Most of these concern calibration, such as sufficiently careful leveling of the optical table and theodolites, measuring the precise position, angle, and illumination distribution of light fixtures (both the arc lamp and fixtures in the track), relatively poor control over the diffuse illumination from the curtains, ceiling, and floor, and the problems inherent in the use of the theodolites for position measurement (time and minimum focus distance).

The safety concerns regarding the arc lamp have already been noted.

There are some unresolved technological problems that we may deal with in the future, such as motorizing the MCP controls, obtaining color images with a parallel-output color camera, and performing real-time motion studies which would require a more sophisticated digitization mechanism. We also have an ongoing schizophrenia concerning the use of metric or English units. While metric units are easier to work with, much of our larger-scale equipment (including the optical table itself) is based on English units.

Finally, there are some optical problems that we may simply have to live with, such as the narrow depth-of-field inherent in small-scale imaging and the movement of the perspective center of a camera when the lens focus is changed.

We believe all of these problems are manageable, and that the CIL will be a successful facility in spite of these annoyances.

6.3 Status and Timetable of Lab Construction

The Calibrated Imaging Lab has been described here as we envision it will be furnished when complete. Much of the equipment is now in place, but the target date for operation to commence is January, 1986. The high-precision camera will not be available until April, 1986.

6.4 Acknowledgements

Takeo Kanade has been involved at all stages in planning the Calibrated Imaging Lab. We are grateful to Stuart J. Smith of the A. and B. Smith Co., Morry Lichter of Lighting Pittsburgh, and Edward J. Lesoon, Jr. of the Asia Carpet and Decorating Co., for their help in obtaining equipment and materials for the lab. Thanks also to Jim Skees and Dennis Royse for their help in making the necessary arrangements at CMU.

END

FILMED

2-86

DTIC

UNCLASSIFIED

Technical Report
distributed by

DEFENSE TECHNICAL INFORMATION CENTER

DTIC
Acquiring Information – Imparting Knowledge

Defense Logistics Agency
Defense Technical Information Center
Cameron Station
Alexandria, Virginia 22304-6145

UNCLASSIFIED

UNCLASSIFIED

NOTICE

We are pleased to supply this document in response to your request.

The acquisition of technical reports, notes, memorandums, etc., is an active, ongoing program at the **Defense Technical Information Center (DTIC)** that depends, in part, on the efforts and interest of users and contributors.

Therefore, if you know of the existence of any significant reports, etc., that are not in the **DTIC** collection, we would appreciate receiving copies or information related to their sources and availability.

The appropriate regulations are Department of Defense Directive 3200.12, DoD Scientific and Technical Information Program; Department of Defense Directive 5230.24, Distribution Statements on Technical Documents *(amended by Secretary of Defense Memorandum, 18 Mar 1984, subject: Control of Unclassified Technology with Military Application)*; American National Standard Institute *(ANSI)* Standard Z39.18, Scientific and Technical Reports: Organization, Preparation, and Production; Department of Defense 5200.1R, Information Security Program Regulation.

Our Acquisition Section, **DTIC-FDAB**, will assist in resolving any questions you may have. Telephone numbers of that office are:

(202) 274-6847, (202) 274-6874 or Autovon **284-6847, 284-6874.**

DO NOT RETURN THIS DOCUMENT TO DTIC

EACH ACTIVITY IS RESPONSIBLE FOR DESTRUCTION OF THIS DOCUMENT ACCORDING TO APPLICABLE REGULATIONS.

UNCLASSIFIED