ERDC/TEC CR-00-4

Topographic Engineering Center

US Army Corps
of Engineers®
Engineer Research and
Development Center

Representation, Modeling, and Recognition of Outdoor Scenes

by Martin A. Fischler and Robert C. Bolles

August 2000

SRI International

Final Report – June 21, 2000

Representation, Modeling, and Recognition of Outdoor Scenes

SRI Project 3388

Contract No. DACA76-92-C-0008

Covering the period: April 1998 through January 2000

Prepared by:

Dr. Martin A. Fischler, Program Director
Dr. Robert C. Bolles, Principal Scientist
Artificial Intelligence Center

Prepared for:

Ms. Lauretta Williams
Code: CETEC-TD-RI
U.S. Army Topographic Engineering Center
7701 Telegraph Road
Alexandria, VA 22315-3864

Approved by:

Dr. C. Raymond Perrault, Director
Artificial Intelligence Center

Table of Contents

PREFACE v

1. OBJECTIVE 1

2. APPROACH 1

3. ACCOMPLISHMENTS (final reporting period) 2

4. DEVELOPMENT OF A SYSTEM FOR AUTOMATED MODELING OF LINEAR STRUCTURES (ESPECIALLY ROADS) IN IMAGES OF THE EARTH'S SURFACE 3

4.1 TECHNICAL SUMMARY: ROAD DELINEATION 3

4.1.1 The Core Problems in Designing an Automated Recognition System 4

4.1.2 Delineation of Roads and Linear Structures 5

4.1.3 A Combinatoric-Based Architecture for LD 5

5. AUTOMATED MODELING OF NATURAL SCENES 7

5.1 TECHNICAL SUMMARY: MODELING NATURAL SCENES 7

6. ERROR-FREE (AUTOMATED) STEREO MATCHING 9

6.1 TECHNICAL SUMMARY: ERROR-FREE (AUTOMATED) STEREO MATCHING 9

6.1.1 The Use of Scene Semantics 10

6.1.2 An Additional Dimension for Appearance-Based Matching 11

7. BIBLIOGRAPHY (Publications describing work performed on this project) 13

APPENDICES 15

PREFACE

This research is sponsored by the Defense Advanced Research Projects Agency (DARPA) and monitored by the U.S. Army Engineering Research and Development Center's (ERDC) Topographic Engineering Center (TEC) under contract DACA76-92-C-0008, titled "Representation, Modeling, and Recognition of Outdoor Scenes." The DARPA Program Manager is **Mr. George Lukes**, and the TEC Contracting Officer's Representative is **Ms. Lauretta Williams**.

REPRESENTATION, MODELING, AND RECOGNITION OF OUTDOOR SCENES

1. OBJECTIVE

The goal in this project was to advance the state-of-the-art in scene modeling and interpretation for autonomous systems that operate in natural/outdoor terrain. In particular, techniques were developed for representing knowledge about complex cultural and natural environments so that a computer vision system can successfully plan, navigate, recognize, and manipulate objects, and answer questions or make decisions relevant to this knowledge.

2. APPROACH

Advances were integrated in four separate technologies to achieve the goal of providing a foundation for the design of highly competent machine vision systems capable of autonomous operation in, and modeling of, the outdoor world.

- Stored knowledge (such as geospatial data and object models, as well as contextual dependencies and interrelationships) is used to overcome inherent weaknesses in the best "self-contained" image-analysis algorithms. This approach is reflected in the prior SRI development of the "CONDOR" and "HUB" systems, and our more recent Automatic Population of Geospatial Databases (APGD)/"BOS" architecture that relies on context, function, and purpose, as well as visually-observed geometric shape, to recognize scene objects.

- Significant progress has been made developing compact and expressive representations for modeling, and ultimately recognizing, objects encountered in the natural world. Computational efficiency, thus, real-time performance, is critically dependent on using effective representations for both reference models and sensed data.

- Global optimization techniques were developed that require reasonable amounts of computation, but produce results not obtainable by local analysis methods. This work has been applied to building volumetric models of objects detected in range data and stereo pairs, as well as for delineation, partitioning, and feature extraction in single images.

- Techniques were developed that are able to simultaneously, or incrementally, exploit multiple views of a scene in compiling a complete scene model. SRI's previously developed epi-polar plane image analysis technique and deformable mesh representations are examples of how multi image collections can be used to construct a geometric scene model that is superior to a sequence of independent stereo reconstructions.

Some of the key ideas underlying this work are that:

- Models are described by objective functions referenced to some appropriate representation; feature extraction is accomplished by finding image structures for which the relevant objective function is optimized. We generally require that the representations we construct be suitable directly-viewable replacements (with respect to the given interpretation task) for

the original image – but require only a small fraction of the original data storage; finding such "reduced representations" are an important step in the solution process.

- Recognition-technique selection and corresponding parameter settings are based on context and confirmed by "built-in" self-evaluation functions.

- We employed a strategy of focusing our development efforts on producing a few highly refined and reliable "core" techniques as the base for implementing a much broader class of feature recognition/extraction methods.

Many of the innovative techniques and ideas that SRI has contributed to the image understanding (IU) program (and to the national machine-vision science and technology base) had their origin in this program. This includes our early work in developing the context-based vision paradigm that led to the RADIUS program, the work on model-based optimization, the deformable-mesh technology, and the linear delineation techniques we transferred to the APGD program (these techniques are now being transferred to the National Imagery and Mapping Agency (NIMA) under separate contract).

After completion of the base funding and development period of this contract, efforts were focused on improving the performance and scope of outdoor scene object recognition techniques. Work on recognizing complex natural and man-made objects (e.g., roads, trees, rocks, and terrain features) is based on a set of ideas and techniques being developed for recognizing complete scene contexts, rather than instances of independent object models. The validity of the approach has been experimentally demonstrated by recognizing and delineating scene objects that cannot be dealt with by conventional methods.

Overall, more than 20 papers have been published describing work on this project.

3. ACCOMPLISHMENTS (FINAL REPORTING PERIOD)

In this final phase of the project, the plan was to focus on research enhancements and algorithm integration to permit testing and technology transfer (especially to DARPA/NIMA-sponsored programs) of some of the most promising recently developed techniques; this work is described in greater detail below and in three appendices.

1. Development of a System for Automated Modeling of Linear Structures (especially roads) in Images of the Earth's Surface

   Work on fully-automated linear delineation in aerial imagery has resulted in what we believe is the most competent algorithm available for this purpose. This algorithm was transferred to and installed in the Radius Common Development Environment (RCDE) for formal evaluation and is now in the process of being transferred to NIMA.

2. Automated Modeling of Natural Scenes

We have developed a set of algorithms for recognizing objects appearing in color photographs of natural outdoor scenes, and for recovering scene geometry without requiring camera calibration or stereo correspondence. While this is an ongoing open-ended task, we can now deal with some recognition problems that had no previously published solutions.

3. Error-Free (Automated) Stereo Matching

An approach has been developed for achieving human-level accuracy in establishing stereo correspondences. The method still remains to be fully implemented, but essentially error-free performances have been demonstrated in recent experiments.

4. DEVELOPMENT OF A SYSTEM FOR AUTOMATED MODELING OF LINEAR STRUCTURES (ESPECIALLY ROADS) IN IMAGES OF THE EARTH'S SURFACE

This seemingly simple task has defied full automation in spite of at least 20 years of effort by a significant number of competent researchers and applied practitioners. The apparent ease of human perception with respect to this task masks the nature of the difficulties. Nevertheless, the techniques developed, largely within this DARPA-supported IU research program, should now allow us to achieve a productivity increase in road modeling of one to two orders of magnitude over the currently employed (largely) manual methods.

4.1 Technical Summary: Road Delineation

In a series of papers presented at IU Workshops, work was described on the detection and extraction of linear features in imaged data: Minimum spanning tree and a novel "network" structure were used as the primary representations. Semantic constraints control the tree/network construction, thus, establish the universe of possible paths (both in our data structures and in the image being analyzed). Characteristics are defined of the linear structures we are looking for as attributes of the branches in the tree/network, and computationally effective methods are provided for finding paths that maximize scores for the desired attributes. Filtering techniques, parameterized by context evaluation procedures (or externally provided information) operate at a number of decision points in the optimization process, and in final acceptance of the selected path(s). Specialized experimental versions of the generic delineation technique were implemented to recognize various types of extended terrain features and navigation obstacles including the skyline, ridgelines, trees, roads, and paths. The problem of finding linear features in aerial images has been of special interest, and has resulted in a major advance in automating the task of modeling roads in the compilation of geospatial databases.

In formal and informal testing on mapping quality aerial images, our completely autonomous delineation module output is typically 90 to 100 percent correct and 80 to 100 percent complete.[1]

Appendix 1 presents the results of work directed at the problem of radically reducing the amount of human effort required to model a road network visible in a collection of images with overlapping coverage of some geographic extent. The primary goal was to develop and

[1] **Correct** = percent of the derived roadmodel that agrees with a human-produced reference model. **Complete** = percent of a human-produced reference model include in the derived model.

demonstrate the technology necessary to enable a factor of 100 reduction over 1996 extraction practice in the time and effort required to produce a road model from aerial and remotely-sensed images for some reasonably broad class of scenes.

In a February 1999 demonstration, an integrated, end-to-end process was shown that produced a 3-D road network model for the McKenna MOUT area at Ft. Benning (using the same images employed in the original benchmark extraction that required 280 minutes of manual effort) that required less than 3 minutes of human effort to edit an automatically produced model -- the automatically produced model itself was 90 percent complete and had a correctness score of 96 percent.

4.1.1 The Core Problems in Designing an Automated Recognition System

Two generic problems must be addressed in any visually-based recognition task.

Problem redefinition. The basic issue is the requirement to express a typically function-oriented description of the object of interest in terms of its visual appearance in an image.

One might expect that an analytic or comprehensive definition of the various features of interest (e.g., roads and buildings) is a necessary first step in the design of the corresponding feature extraction algorithms/systems. We assert that from a practical standpoint, it is impossible to provide a comprehensive computational definition of something with instances as geometrically diverse and complex as a "road" or a "building."

Dictionary definitions of roads and buildings are primarily concerned with their use, rather than their geometric structure or appearance. Even if it were possible to provide the desired definitions, there will always be a significant number of ambiguous cases. For example, at what point does a road under construction or a very long driveway become a road, or long continuous shoulders become an extra highway lane? If a small segment of a road is not visible in an image, should the modeling system fill it in even though it could be due to an actual gap in the continuous road surface? If a vehicle can easily cross from one road to another adjacent road (say over an open divider strip), should we insert an intersection at such allocation even though it is "illegal" to cross over?

A feature extraction algorithm embodies an implied computational definition of the feature it is intended to model. The algorithm designer usually bases his design on (1) requiring the visible/measurable presence of certain structures or conditions (e.g., a road must exceed some minimum length, width, and lie on the earth's surface); (2) requiring the absence of other structures or conditions (e.g., a road cannot radically change direction or width very often); and (3) assumptions about the scene being modeled (e.g., roads in San Francisco can be assumed to be paved rather than dirt).

The ultimate user of the model probably has in mind a functional (dictionary style) definition of the features in the model; for example, a road is a physical structure that facilitates the movement of vehicles, and indeed is used for that purpose. Human image analysts use both types of definitions, but the key point is that there is no single common definition that can be used as the ultimate basis for deciding whether a model is correct or incorrect. Even if we adopt the end

user's definition, we still have the problem that an image taken in isolation is rarely able to provide all of the information needed to establish if such a definition is (or is not) satisfied. Thus, the first problem to be solved is to provide a computational redefinition of the nominally given problem that produces answers consistent with the expectations of a potential user.

Design of a computationally feasible solution for "real-world" problems. Most recognition problems, treated in their full generality, are computationally intractable; it can easily be shown that road delineation also has this characteristic. A critical issue in the design of a recognition system that has acceptable performance on reasonably-sized "real-world" problems is how to make appropriate tradeoffs between computational complexity and the use of approximations and/or accepting a limited amount of error. A key aspect of our design was to find a way to reduce the image information content by more than 2 orders of magnitude in the first major processing step, while still retaining the information essential to obtain a good approximation to the desired solution. Another, more practical, measure taken to control the computational requirement is to restrict all algorithms to be O(nlogn) in their theoretical complexity.

4.1.2 Delineation of Roads and Linear Structures

Because of occlusions and background clutter (i.e., objects appearing in an image that are of no interest other than that they mimic the objects we are searching for), there is generally no simple way to partition the image into curves corresponding to coherent line-like objects that are complete and have no contamination by extraneous background content. If we take a simple generate-and-test approach, an image with as few as 20 curve points would be computationally impractical to process because of the factorial growth in the possible number of curves connecting the points. What is implied by the above considerations is that a single-step solution is probably not attainable; we must perform a sequence of grouping, filtering, and information-reduction steps to eliminate unlikely candidates as early in the process as possible, and then make the final selection on a greatly simplified reduction of the originally presented data.

4.1.3 A Combinatoric-Based Architecture for LD

In general, we must address two subproblems: (1) selecting/partitioning the actual path points from the set of potential path points, and (2) sequencing the selected path points (see Appendix 2).

To solve the unconstrained delineation problem we must strictly limit the number of points that can be arbitrarily sequenced, or we must limit the number of choices that are the possible successors of any given point, or use some combination of the two preceding constraints. We have observed that we can usually find dense path segments and place perceptual and/or application-domain-related constraints on linking possibilities for these dense segments. To the extent that most of the path points are already sequenced as members of the detected segments, and it is only the segments that must be sequenced, and even here there are only a few linking alternatives for each of the segments, we can solve the sequencing problem even though it is formally intractable.

The overall approach is to:

1. Detect potential road points in black/white panchromatic images and construct a binary representation that retains the perceptually obvious linear structure. This information reduction step is critical in allowing us to employ very efficient and expressive graph-theoretic methods to solve the delineation problem.

2. Assemble potential path points into dense segments by using a fast Minimum Spanning Tree (MST) algorithm (although the MST does not actually assure the densest connectivity, it usually provides a very good approximation to this condition). Recover the longest smooth segments -- consistent with generic perceptual connectivity criteria -- which can be extracted from the forest of trees generated in this step. The input to this step is the binarized image; the output is a list of disjoint (potential road) segments.

   In a typical road delineation problem we started with a 768X638-pixel image (489,984 points) and constructed a binarized version with 55,480 potential road points. As a result of step 2, we extracted 340 segments containing 21,255 points.

3. Repartition and semantically filter the initial collection of paths to eliminate perceptual and semantic linking mistakes and irrelevant paths introduced or retained by the limited flexibility of the MST algorithm/representation and the generic parsing process.

4. Use a recently-developed linking technique and representation schema, capable of expressing arbitrary perceptual and semantic constraints, to imply a network of paths that is likely to include the road network to be modeled.

5. Refine the 3-D geometric shape and location of the delineated road segments.

6. Employ contextual and semantic knowledge to produce a final enhancement of the fully automatic phase of road model construction. Eliminate obvious errors and flag dubious entries for interactive editing.

Finding roads is decomposed into the problems listed above because the more direct approach of selecting and evaluating arbitrary paths is computationally infeasible: the combinatorics of indiscriminant search makes it simpler to first solve the more general problem of finding (generic) linear structures, and then filtering out roads, than to examine all possible paths in an image to find the roads directly.

The detection task must address the problem that it is often impossible to distinguish roads from other natural or man-made structures at a local level. For example, if we look at an image through a small peephole, objects such as a section of a river, or a parking lot, or the roof of a rectangular building can all have an appearance similar to that of a small stretch of road. It is also the case that tunnels, trees, buildings, clouds, and so forth, can occlude some section of a road, causing an apparent break in road continuity.

The generic linking must address a number of problems. Given that the detection task is not error free (i.e., we expect to have both false alarms and misses) the implied connectivity is ambiguous or possibly incorrect. There also is the computational problem of actually linking the

individually-detected road points into continuous segments and connected networks that represent the road structures for which we are searching.

Because of possible errors in assumed connectivity, and because roads and other linear structures may be intermixed at the level of generic linking (e.g., some linked path could be a composite of a road segment attached to some other non-road object), subsequent steps in the road delineation process must permit some relinking as well as the recognition of road versus non-road segments in the networks returned by the generic linker.

Our delineation system as currently implemented has effectively addressed each of the above problems and is capable of making a major contribution to fully automated road modeling from aerial images.

5. AUTOMATED MODELING OF NATURAL SCENES

There is a huge hole in the ability to perform the recognition task when the objects of interest cannot effectively be identified by their explicit geometric shape, or by some directly measurable attribute (e.g., gross size, color, speed of movement). This means that we cannot deal effectively with most natural objects and features of the outdoor world -- vegetation, rocks, water, sky, land-forms, and terrain features.

For some tactical applications, it may be possible (actually necessary with existing technology) to focus on the man-made objects of primary interest, and treat the natural objects as background "noise" rather than as context for helping to understand/model the overall scene; however, for many other applications (e.g., synthetic environments), we must be able to produce realistic 3-D reconstructions of these natural backgrounds.

There is typically no need to construct accurate geometric models of (say) the vegetation, but unless we can identify, at some less detailed level, the visually prominent natural objects, we have no realistic way of "approximating" their appearance in a rendering of the scene.

The problem of modeling natural scenes was addressed as one of the earlier tasks in this project. The goal was to be able to take one or more images -- preferably color, and possibly uncalibrated - and recover the salient natural features and qualitative geometric structure of the actual scene. The recovered model should look like a realistic rendering composed with some artistic license [M.A. Fischler, "Robotic Vision: Sketching Natural Scenes," Proc. ARPA Image Understanding Workshop, Palm Springs, CA., February 1996]. The most recent focus for this work is discussed below, where the ability to classify material surfaces is shown to play a critical role in effective stereo modeling.

5.1 Technical Summary: Modeling Natural Scenes

The goal in this task was to be able -- using completely automatic techniques -- to recognize natural and (some) man-made objects and terrain features in the context of creating an overall qualitative scene model or sketch. We are not only interested in recognizing and delineating isolated objects, but want to describe and exploit their interrelationships. Objects of interest

include grass, brush, trees, rocks, ridgelines/skyline, snow, water, shadows, fences, poles, holes/ditches, and roads/paths.

To the naive eye, usually, the sky is blue, vegetation green, the earth gray/red/brown, water blue/green, and so forth. Is it possible to take a real color image, and on a local (or even pixel-level) basis, produce a "false" color image with a few colors (say 4 to16), each color corresponding to a specified semantic category, and the false-color image itself a recognizable replacement for the original -- not only with respect to semantic labels, but also allowing recovery of gross terrain geometry? If such recoloring is indeed possible, as our experiments seem to imply, the implications are quite profound. Such an easily-derived explicit representation (the Color-Sketch) could provide a way for a simple organism (animal or animate -- without conventional language machinery, higher-level reasoning, or sophisticated mathematical manipulation) to base immediate (visually-guided) behavior on semantic considerations.

In attempting to design a vision system for a robotic device (even a vision system limited to supporting the task of outdoor navigation) and encountering a host of refractory problems, one cannot help wondering how simple biological organisms can, seemingly, perform this task so well. While the nominal concern is ultimately to support a full range of interactions of the robot with its environment, a more achievable initial objective is to consider only those aspects of visual interpretation required for local navigation. The semantic vocabulary could be as simple as go/no-go directions open to the robot. It is more important to recognize such functionally meaningful image-point-attribute distinctions as solid/deformable, flat/raised, close/distant, than specifically recognizing that something is a tree rather than a rock. Nearby objects should be given more attention (with respect to positional accuracy and semantic resolution) than distant ones, which can be dealt with again at a later time if necessary. A subjective (viewer-centered) model (e.g., an iconic overlay of the image) that can be used for reactive behavior (as noted above) turns out to be relatively easy to derive as compared to an objective model (e.g., a symbolic labeling of the partitioned scene) that is required for long-range planning. To the extent that the sensing modalities are available (e.g., stereo, motion, color, and polarization) they can pay very high dividends in the simplification of the interpretation task over what can be accomplished with single black and white images.

The Scene-Sketch has direct utility for reactive robotic navigation since its overlays of the scene allow the robot to quickly determine the likely presence of raised objects, flat navigable areas, and surface material in any view direction. This information is available in qualitative form even without the availability of explicit depth overlays (say, from stereo) or the need for explicit partitioning. A significant number of pixels with the same semantic or geometric label in a particular view direction tells the robot what it is likely to encounter if it moves in that direction. The vertical position (y-coordinate) of the first (smallest y-coordinate) pixel in a coherent sequence of identically labeled pixels provides an estimate of the distance to the corresponding object/region.

Since the Scene-Sketch is qualitative, and its vocabulary is limited, its appropriate use beyond reactive navigation is as input to higher-level analysis processes. For example, since any non-sky pixel in the color sketch located above the skyline (in the line-sketch) can be assumed to be a

pole or raised vegetation (a tree or a large bush), we can easily extend the semantic vocabulary of the primitive Scene-Sketch to include these additional objects and detect them with relatively simple algorithmic techniques. We also can invoke simple rules to check physical consistency, for example a pixel-labeled water cannot (correctly) lie vertically above a pixel-labeled sky.

6. ERROR-FREE (AUTOMATED) STEREO MATCHING

The key observation here was that many of the assumptions underlying automated stereo modeling (and useful for conventional aerial mapping) do not hold for ground-level scenes. Specular reflection from water, glass windows, metallic surfaces, and so forth, violates the usual assumption of Lambertian reflectance. The nominal assumption that we are dealing with "mostly" continuous surfaces fail for the sky region and for most vegetated regions (e.g., nearby tall grass or leaves). The assumption that highly textured surfaces provide reliable match candidates fails in the case of man-made repetitive textures (e.g., the windows on the wall of a building) rather than the assumed random textures. Occlusion is a very common, rather than rare, occurrence and must be explicitly modeled. It is thus apparent that stereo in outdoor ground-level scenes can be successful only when the surfaces and objects being modeled can be recognized with respect to a few (~10 to 15) pervasive material categories (including sky, water, raised vegetation, man-made surface, rock, bare earth, transparent/translucent material). The categorization does not have to be exhaustive nor exclusive -- we have been able to successfully apply our previously discussed color-based classification results (as well as employing a new learning-based classifier developed at Stanford) to meet our needs.

The overall approach was to devise a technique that can find, and "guarantee" the absolute correctness of the correspondences of, at least a few hundred points in two or more images of some given scene. It appears that this can be done, at least with an error rate of less than 1 to 2 percent, and we believe we can use the approach recursively in successively narrower matching contexts to obtain human-level performance in dense stereo matching. Progress on this project was reported in a paper published in the IUW98 proceedings [ref: J.Z. Wang and M.A. Fischler, "Visual similarity, judgmental certainty, and stereo correspondence"] and the final test and evaluation results are included in Appendix 3. It is interesting to note that most of the current work on urban modeling of buildings is based on using a large number of accurately-registered images in order to "cancel" out the existence of vegetation and other "problem" surfaces, rather than including them in the final model. Our approach is to use two or three images and produce a complete model for the imaged data.

6.1 Technical Summary: Error-Free (Automated) Stereo Matching

Although part of a more general concern for how to recognize the same physical location (or object) in two or more views of a given (natural) scene, we focused on a "simplest" version of this problem: how a machine can establish stereo correspondence under conditions similar to those encountered by the human visual system (HVS) - in particular, from two images acquired from the same vertically-oriented camera closely positioned at two locations in space and time. Using the HVS as an existence proof, it appears that "almost perfect" recovery of qualitative scene geometry is possible. This appears to imply almost error-free matching of corresponding points in the two images as required for stereo triangulation, and since we must be capable of

dealing with arbitrarily selected natural scenes, any prior knowledge of the local scene geometry is ruled out.

We argue that in order to achieve (essentially) error-free performance, a number of deficiencies in the conventional assumptions and computational approach to (two-image) stereo matching must be corrected:

1. It is generally assumed that two image-based projective-geometric reconstructions can be accomplished *solely* by appearance-based dense matching under geometric constraints that establish the epi-polar relationship of the two images. (Eucledian camera models are preferred, and exploited when available, but do not add an extra dimension to the matching task -- however, they are necessary for the final recovery of a 3-D Euclidean model of the scene.)

2. It is almost universally assumed that the scene is mostly composed of continuous (solid) surfaces with Lambertian reflectance properties.

3. It is almost universally believed that knowledge and use of scene semantics is neither necessary nor feasible for stereo matching and geometric reconstruction.

It does not take deep reasoning to see that there are serious problems with all of the above assumptions. It is somewhat more problematic to provide practical alternatives. We summarize our contribution with respect to how information from three nominally independent and informed knowledge sources can be exploited to deal with some of the obvious shortcomings of the current stereo paradigm.

6.1.1 The Use of Scene Semantics

Images of natural scenes are generally unmatchable over a considerable portion of their extent. In addition to relative occlusions (between the two images) due to raised objects, we have the problems of non-Lambertian surfaces (e.g., water, snow, clouds, granite . . .), non-solid surfaces (e.g., clouds, water, grass, foliage), and transparent media (e.g., sky, water). To the extent that we can partition natural scenes into a few semantic categories, we can avoid errors due to finding matches in "unmatchable" regions, or matches between points/surfaces that have different semantic identities. Even when we are attempting to find matches between locations in the two images with corresponding semantic partitions, we still gain by not mixing the statistics associated with the different semantic types.

It is not hard to achieve a useful level of semantic partitioning. Credible results are shown from fairly simple currently available algorithms operating on colored imagery. The errors are not serious; they simply reduce the possible number of asserted correct matches -- but we already know that much of the reconstruction will require some form of interpolation from a core set of correct (i.e., highly reliable) matches.

6.1.2 An Additional Dimension for Appearance-Based Matching

Conventional stereo-matching techniques can be viewed as representing a point in an image as a vector composed of the image intensities in a square (surface patch) centered on the given point. The size of the patch is a tradeoff between a number of factors -- the smaller the size, the better the resolution of the recovered geometric model, but the more likely the possibility of allowing an incorrect match. There also is a problem in allowing the size of the patch to become too large, since the underlying assumption that the patch represents a planar approximation to a coherent piece of the viewed surface, at a fixed depth, will eventually be violated as the patch size is increased. Matching is usually accomplished by finding the "most similar" vector in the conjugate image that satisfies a number of constraints -- especially, that the conjugate point lies on the epi-polar line of the given point if an epi-polar transform is available.

A critical problem is what do we mean by "most similar." We need to define a similarity metric that is invariant to the known artifacts introduced by the imaging process, or remove these artifacts by appropriately modifying the representation vectors -- in practice, both of these approaches to removing artifices are employed. There also is the problem that the components of the representation vector are generally not equally important in establishing correct correspondence and must be differentially weighted, or their dimensionality reduced to remove correlated information. A large number of similarity metrics has been proposed for dealing with the above problems; however, one of three variants of a Euclidean-distance metric in the vector-representation space is almost universally employed for stereo matching.

The three variants are:

1. The Direct Euclidean-metric (DEM).

2. The Correlation-Metric (CM) which normalizes the DEM for an unknown linear transform (bias and gain, that is, the mean value and range of intensities) between the intensities of the two images of the conjugate pair. The CM measures the angle between two vectors as their similarity score.

3. The Mahalanobis-distance (MD), which eliminates the effect of linearly correlated information between the components of each vector, and then measures the Euclidean distance between the resulting vectors.

There is no clear dominance of any one of the above metrics over the others. The CM is most widely used since it is computationally simple and removes a common source of error (the use of absolute rather than relative intensities); however, it throws away information that would be useful if we could be sure it was not contaminated. The MD will provide the most desirable form of normalization if we can be sure that the sample statistics it uses to compute the normalizations are not distorted by incorrectly matched conjugate pairs, or by pairs from different semantic categories with significantly different intervariable correlations.

In our matching formulation, we use the MD and assume each conjugate pair represents two representative samples drawn from the same distribution, and that all conjugate pairs -- in the

same semantic partition -- are identically distributed with different means. The covariance matrix needed for the MD normalization is the average of the matrices for all the matched pairs in the given semantic category. If we combined all the matched pairs into the computation of a (single) global covariance matrix, the result would be a less consistent distance metric, and if there also were incorrect matches, the utility of the MD metric could be lost or even degraded below that obtainable
from one of the other computationally simpler metrics. We also could remove the global mean from each vector individually (as in the case of the CM) and compute the combined covariance matrix for the entire population, but again, it is not clear that this result would be worth the effort over just using the CM or DEM.

The similarity metric, by itself, does not provide assurance of a correct match. We also must be certain that there is only one conjugate point that is closely similar to a given point before the pair is accepted as a correct match. This requirement is usually dealt with in a rather ad-hoc way in most stereo matching systems; a number of mechanisms have been employed, including the following:

1. Consider the two top candidates in Image2 as conjugates for a given point in Image1. Accept the match only if the difference in similarity scores between the two pairs exceeds some specified threshold (e.g., 1 or 2 standard deviations in the distance distribution of a given point from its correct conjugate).

2. Find the point (Q2) in Image2 that is the most similar to a given point (Q1) in Image1. Accept the pairing only if a search for the point most similar to Q2 in Image1 produces the point Q1.

3. Only match points that have a high intensity variance over their vector components in the hope that they will be distinct enough to produce only one potential conjugate (such points are called "interest points").

We view the requirement for an effective and principled uniqueness metric as important as the need for a similarity metric, and in a significant departure from existing stereo matching methodology, we have introduced such a metric (see Appendix 3). One key difference between what we propose and the existing adjuncts to the similarity evaluation is that we filter potential conjugate pairs before they are able to contaminate the statistics required for the similarity metric to operate effectively. By examining both images for points similar to a given point, we often find evidence for ambiguity in the image containing the point while such evidence is masked by occlusion in the conjugate image.

In Appendix 3, we provide a more complete description of our approach and an extensive tabulation of experimental results to quantify our claim of essentially error-free performance.

7. BIBLIOGRAPHY (Publications describing work performed on this project)

P. Fua, "Fast, Accurate, and Consistent Modeling of Drainage and Surrounding Terrain," IJCV 26(3):215-234, March 1998.

P. Fua and Y.G. Leclerc, "Combining Stereo, Shading, and Geometric Constraints for Surface Reconstruction from Multiple Views," Technical Conference Geometric Methods in Computer Vision II of SPIE Symposium, San Diego, CA, July 1993.

P. Fua and Y.G. Leclerc, "Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading," IJCV, 1994.

P. Fua and Y.G. Leclerc, "Using 3-Dimensional Meshes to Combine Image-Based and Geometry-Based Constraints," ECCV, Stockholm, Sweden, May 1994.

P. Fua and Y.G. Leclerc, "Registration without Correspondences," CVPR, Seattle, WA, pp. 121-128, June 1994.

P. Fua and Y.G. Leclerc, "A Unified Framework to Recover 3-D Surfaces by Combining Image-Based and Externally-Supplied Constraints," Proc. ARPA Image Understanding Workshop, Monterey, CA., November 1994.

P. Fua and Y.G. Leclerc, "Image Registration without Explicit Point Correspondences," Proc. ARPA Image Understanding Workshop, Monterey, CA., November 1994.

P. Fua, "Surface Reconstruction Using 3-D Meshes and Particle Systems," Third International Workshop on High Precision Navigation, Stuttgart, Germany, April 1995.

P. Fua, "Reconstructing Complex Surfaces from Multiple Stereo Views" (Submitted to ICCV, Boston, MA, June 1995).

M.A. Fischler and H.C. Wolf, "Saliency Detection and Partitioning Planar Curves," Proc. Image Understanding Workshop, Washington D.C., pp. 917-931, April 1993.

M.A. Fischler and H.C. Wolf, "Locating Perceptually Salient Points on Planar Curves," IEEE-PAMI, vol. 16(2):1-17, February 1994.

M.A. Fischler, "The Perception of Linear Structure: A Generic Linker," Proc. ARPA Image Understanding Workshop, Monterey, CA., November 1994.

M.A. Fischler, "Robotic Vision: Sketching Natural Scenes," Proc. ARPA Image Understanding Workshop, Palm Springs, CA, February 1996.

M.A. Fischler, "Finding the Perceptually Obvious Path," ARPA Image Understanding Workshop, May 1997.

M.A. Fischler and Aaron J. Heller, "Automated Techniques for Road Network Modeling," Proc. DARPA Image Understanding Workshop, 1998.

M.A. Fischler and Robert C. Bolles, "Evaluation of a Road-Centerline Data Model," Proc. DARPA Image Understanding Workshop, 1998.

M.A. Fischler, R.C. Bolles, A.J. Heller, and C.I. Connolly, "An Integrated Feasibility Demonstration for Automatic Population of Geospatial Databases," Proc. DARPA Image Understanding Workshop, 1998.

Y.G. Leclerc and M.A. Fischler, "An Optimization-Based Approach to the Interpretation of Single Line Drawings as 3-D Wire Frames," Int. J. Computer Vision, 9(2):113-136, 1992.

T. Luong, "Sketching Natural Terrain from Uncalibrated Imagery," ARPA Image Understanding Workshop, May 1997.

W. Neuenschwander, P. Fua, G. Szekely, and O. Kubler, "Initializing Snakes," CVPR, Seattle, WA, June 1994.

W. Neuenschwander, P. Fua, G. Szekely, and O. Kubler, "Using Boundary Conditions to Improve Snake Convergence," ICPR, Jerusalem, Israel, October 1994.

T.M. Strat and M.A. Fischler, "The Role of Context in Computer Vision," Proc. Workshop on Context-Based Vision, Cambridge, MA, June 1995.

J.Z. Wang and M.A. Fischler, "Visual Similarity, Judgmental Certainty, and Stereo Correspondence," Proc. DARPA Image Understanding Workshop, 1998.

APPENDICES

APPENDIX 1: Automated Techniques for Road Network Modeling

APPENDIX 2: Automated Road Modeling: Constructing the Road Graph

APPENDIX 3: Visual Similarity, Judgmental Certainty, and Stereo Correspondence

APPENDIX 1:

Automated Techniques for Road Network Modeling

Automated Techniques for Road Network Modeling *

Martin A. Fischler and Aaron J. Heller
Artificial Intelligence Center, SRI International
333 Ravenswood Ave., Menlo Park, CA 94025 USA
E-MAIL: {fischler,heller}@ai.sri.com

Abstract

In this paper, we present the results of work directed at the problem of radically reducing the amount of human effort required to model a road network visible in a collection of images with overlapping coverage of some geographic extent.

1 Introduction

Our primary goal in the first year of this APGD task was to develop and demonstrate the technology necessary to enable an order of magnitude (factor of 10) reduction over 1996 extraction practice in the time and effort required to produce a road model from aerial and remote-sensed images for some reasonably broad class of scenes. It was agreed early in the program that initial efforts would focus on the Ft. Benning McKenna Military Operations in Urban Terrain (MOUT) facility and surrounding area. This area contains approximately 20 km of roads over an area of 6.5 km^2 and is covered by 44 frames of 1:5000-scale panchromatic mapping photography.

The baseline performance benchmark of 280 minutes was established by the extraction task for the McKenna MOUT area carried out by a professional cartographer using a digital stereo photogrammetric workstation (DSPW) running SocetSet software [Goddard, 1996]. This benchmark was part of the 1996 High-Resolution Model Extraction study sponsored by the US Army Topographic Engineering Center (USATEC).

In a formal demonstration held at SRI on 22 April 1998, we showed an integrated, end-to-end process (Figure 1) that produced a 3-D road network model for the McKenna MOUT area at Ft. Benning (using the same images employed in the original benchmark extraction) that required approximately 25 minutes of human effort to edit an automatically produced model – the automatically produced model was 86% complete and had a correctness score of 90%. We also informally demonstrated the ability of the automated segment of the system to model roads visible in five-meter resolution National Imagery and Mapping Agency (NIMA) Controlled Image Base (CIB) data.

In the remainder of this report we will describe the nature of the technical problems we had to address and the approach we developed to reach our first year goal.

2 Technical Background

The task of automatically recognizing and extracting a given class of features from unconstrained aerial images, at anything approaching a human level of performance, remains an unsolved problem in general. We have made significant progress in automatic techniques for recovering scene ge-

*This work was sponsored by the Defense Advanced Research Projects Agency under contract NMA100-97-C-1004 monitored by the National Imagery and Mapping Agency, Reston, VA, and contract DACA76-92-C-0008 Monitored by The U.S. Army Topographic Engineering Center, Alexandria, VA. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the United States Government, or SRI International.

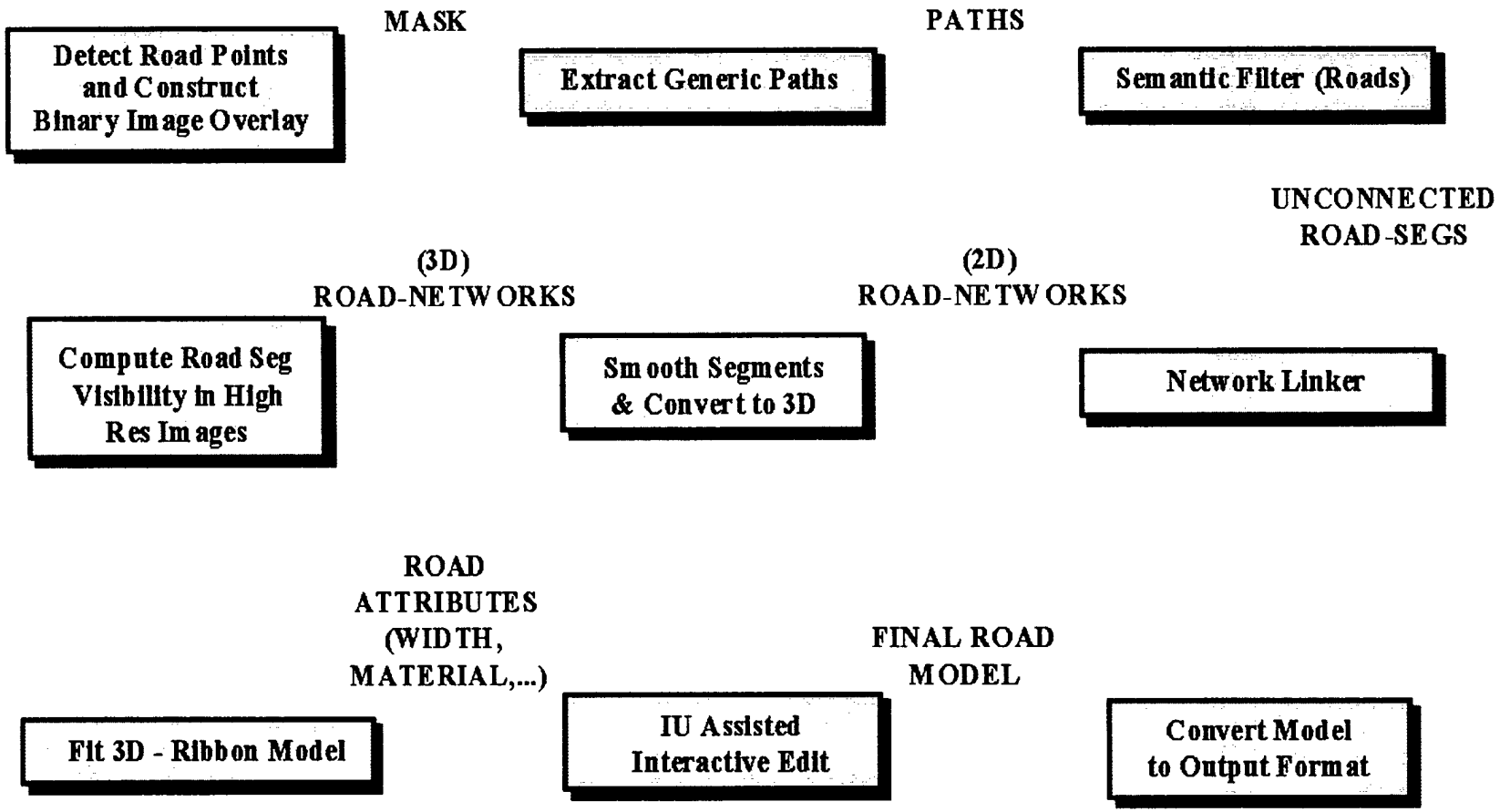


Figure 1: Block diagram of the end-to-end Road Modeling System.

ometry and can automatically recognize objects that have explicit geometric descriptions. However, except for the case where a special sensor measurement is enough to do much of the job (e.g., recognizing bodies of water in infrared imagery), the only approach that works in general is to narrow the context to the point that only a few alternatives are possible.

For example, if we are looking for an object that can be found at a known geographic location, such as a submarine in a specific pen, then we can usually determine if the object of interest is present or not. If the submarine is away from its pen, say visible on the water surface but disguised to look like a fishing trawler, we would have very little hope of finding it using current automated image examination techniques.

2.1 The Core Problems in Designing an Automated Recognition System

There are three primary problems that must be addressed in a visually based recognition task:

1. Problem redefinition. The basic issue here is the requirement to express a typically function-oriented description of the object of interest in terms of its visual appearance in an image.

One might expect that an analytic or comprehensive definition of the various features of interest (e.g., roads and buildings) is a necessary first step in the design of the corresponding feature extraction algorithms/systems. We assert that from a practical standpoint, it is impossible to provide a comprehensive computational definition of something with instances as geometrically diverse and complex as a "road" or a "building."

Dictionary definitions of roads and buildings are primarily concerned with their use, rather than their geometric structure or appearance. Even if it were possible to provide the desired definitions, there will always be a significant number of ambiguous cases. For example, at what point does a road under construction or a very long driveway become a road, or a long continuous shoulder become an extra highway-lane? If a very small segment of a road is not visible in an image, should the modeling system fill it in even though it could be due to an actual gap in the continuous road surface? If a vehicle can easily cross from one road to another adjacent road (say over an open divider strip), should we insert an intersection at such a location even though it is "illegal" to cross over?

A feature extraction algorithm embodies an implied computational definition of the feature it is intended to model. The algorithm designer usually bases his design on (1) requiring the visible/measurable presence of certain structures or conditions (e.g., a road must exceed some minimum length, width, and lie on the earth's surface); (2) requiring the absence of other structures or conditions (e.g., a road can't radically change direction or width very often); and (3) assumptions about the scene being modeled (e.g., roads in San Francisco can be assumed to be paved rather than dirt roads).

The ultimate user of the model probably has in mind a use-based (dictionary style) definition of the features in the model – e.g., a road is a physical structure that facilitates the movement of vehicles, and indeed, is used for that purpose. Human image analysts use both types of definitions, but the key point is that there is no single common definition that can be used as the ultimate basis for deciding whether a model is correct or incorrect. Even if we adopt the end user's definition, we still have the problem that an image taken in isolation is rarely able to provides all of information needed to establish if such a definition is (or is not) satisfied.

Thus, the first problem to be solved is to provide a computational redefinition of the nominally given problem that produces answers consistent with the expectations of a potential user.

2. Design of a computationally feasible solution for "real world" problems. Most recognition problems, treated in their full generality, are computationally intractable; in a following section we show that road delineation also has this characteristic. A critical issue in the design of a recognition system that has acceptable performance on reasonably-sized "real world" problems is how to make appropriate tradeoffs between computational complexity and the use of approximations and/or accepting a limited amount of error. A key as-

pect of our design was to find a way to reduce the image information content by more than two orders of magnitude in the first major processing step, while still retaining the information essential to obtain a good approximation to the desired solution (e.g., see figures 2 and 3). Another, more practical, measure taken to control the computational requirement is to restrict all algorithms to be $O(n\dot{l}ogn)$ or $O(n)$ in their theoretical complexity.

3. Self-evaluation or knowing when you have the correct answer. A central theme of the APGD IFD effort is to achieve system robustness and reliability. An algorithm that is robust and predictable under narrow but well-understood and documented conditions is much more valuable as a system component than a second algorithm that scores very well in a given benchmark evaluation, but for which the designer is unable to provide performance charactarizations or guidelines for its use in different contexts, and which cannot evaluate its own performance.

The key to robustness is the ability of an algorithm to know when it has produced a questionable answer (correctness can never be assured). A good theoretical solution to this problem is still not available in general, but we have made some important practical progress in the case of road modeling by finding sets of constraints on a valid solution that can be progressively tightened to retain only the best candidate models. This capability allows us to select an appropriate "operating point" for the algorithm; that is, we can trade missed detections (false negatives) for false alarms (false positives) depending on system requirements.

2.2 State of the Art

The problem of automatically delineating roads in aerial images has been under study by computer scientists for over 20 years (e.g., early work includes [Quam, 1978, Nevatia and Babu, 1978, Bolles *et al.*, 1979, Fischler *et al.*, 1981]). Numerous algorithms have been developed to date although almost all the linear delineation algorithms are "trackers" in the sense that they follow a single path. They generally require that they be given a start point, a direction, and width information. The main distinction between these trackers is whether or not they depend on internal detail: at high resolution, a linear feature, such as a road, is a ribbon with internal structure rather than just a thick line. Most trackers are variants on two basic themes, sequential line/edge/intensity-feature followers [Quam, 1978, McKeown and Denlinger, 1988], or "path-cost" optimizers [Fischler *et al.*, 1981, Fua and Leclerc, 1990, Iverson, 1997].

The sequential followers search locally for the continuation of a partially formed track; they can be very fast and effective in following a clearly visible, continuous, isolated track, but under more difficult conditions, they generally have trouble telling when they have made a mistake, as well as recovering from a mistake.

The path-cost optimizers come in a number of varieties, but in theory, they are able to select the least-cost path connecting two specified points in an image. The cost of a path is typically taken as the sum of the costs assigned to the individual pixels traversed by the path and a cost assigned to local path geometry, curvature for example, or to some relationship between the attributes of pairs of successive path pixels. The global optimizers always do what they are told–produce the lowest cost solution–but in practice this is not necessarily the desired answer. For example, when a weakly visible road parallels a nearby clearly visible one, it is difficult to delineate the weak road since the tracker prefers to jump over to the strong road where the costs are lower.

With the exception of this effort, and earlier work done at SRI on automated extraction of complete road networks [Fischler and Wolf, 1983, Fischler, 1994, Fischler, 1997], there are very few systems (described in the open literature) that can reliably extract complete road networks from aerial images without an externally supplied image-specific initialization or guiding sketch.

In the remainder of this paper we will describe the specialization of the BOS architecture, under the control of the road-modeling Feature Extraction Manager (FEM) to perform the road modeling task (Figure 1). Five distinct linear-delineation algorithms are invoked: (a) a graph-theoretic network-delineator to "quickly" extract (global) road net-

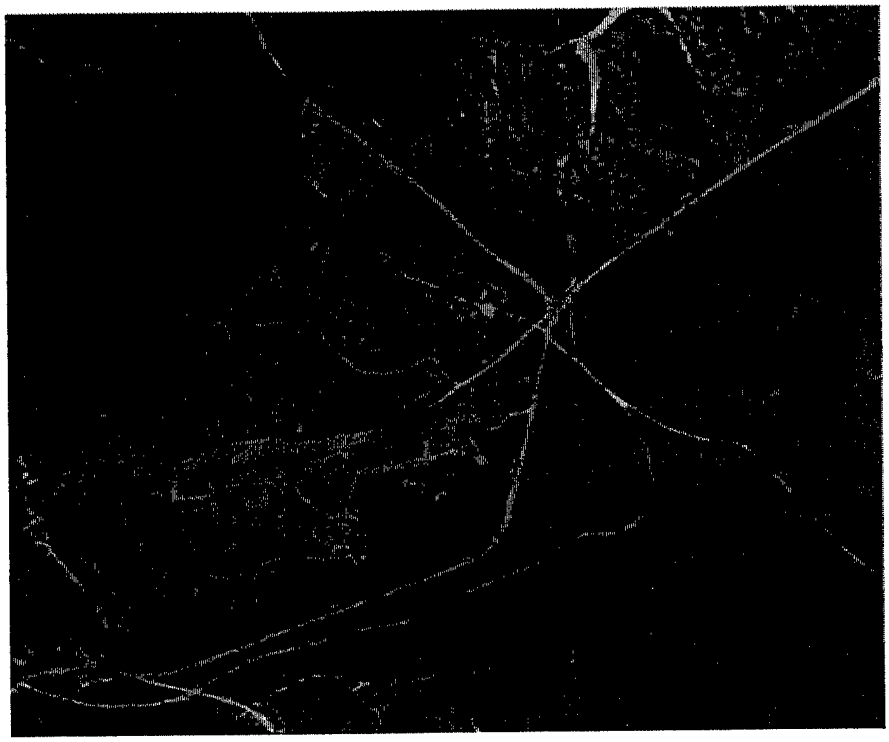


Figure 2: Ft. Benning McKenna MOUT orthomosaic.

Figure 3: Linear structure mask created from Ft. Benning orthomosaic.

work topology from low resolution synoptic 2-D imagery (e.g., an orthophoto of the complete site being modeled), (b) a pair of 3-D snaking algorithms, "cued" by the "sketch" produced by the road-network-delineator, to refine the the local road-geometry and produce a 3-D ribbon using (when available) multiple high-resolution images, and in a final editing step (c) a collection of interactive path-delineators (including a dynamic-programming path-optimizer and a correlation-based path-follower) to refine the final result and produce a product that satisfies application-specific standards and constraints.

3 System Architecture

The high-level architecture of the system involves three distinct processes:

Low-resolution road detection and recognition. A completely automatic process (called LD) that is capable of accepting even a single aerial image that is assumed to satisfy a few assumptions and return a delineation and topological description of the network of visible roads, which we refer to as the *road centerline model.* The result produced by LD is computed at a nominal ground resolution of two to eight meters per pixel, regardless of the availability of higher resolution images.

High-resolution, 3-D refinement and attribution. When high resolution imagery (less than 1 meters per pixel) is available, the result produced by LD can be used to "index into" a collection of such images to allow a more detailed, context-based analysis of questionable decisions made by LD, more accurate positioning of the road bed, and determination of road attributes including road width, surface material type, along- and across-road gradient, and so forth. The output of this fully-automatic process is an attributed road-network model. (This High Resolution Analysis process is called HRA.)

IU-assisted interactive editing and review. It is necessary to provide an editing capability as a buffer between the special needs of different practical mapping and scene modeling applications, and the capabilities of the rather general fully-automatic processes listed above. The third component of the complete system is a highly efficient interactive editor that is able to employ a significant inventory of automated tools under human control.

The entire road modeling system (still in an intermediate unoptimized stage of development) can nominally model 2 km of road per minute in the fully automatic first two phases, and produce a professional level edited product at the rate of 0.5-1.0 km road per minute. It has been implemented as a layered system on top of the RCDE, a highly-integrated 3-D cartographic modeling system that runs on SGI and Sun Microsystems workstations [Heller and Quam, 1997].

4 Delineation of Roads and Linear Structures

Because of occlusions and background clutter (i.e., objects appearing in an image that are of no interest other than that they mimic the objects we are searching for), there is generally no simple way to partition the image into curves corresponding to coherent line-like objects that are complete and have no contamination by extraneous background content. If we take a very simple generate-and-test approach, an image with as few as 20 curve-points would be computationally impractical to process due to the factorial growth in the possible number of curves connecting the points. What is implied by the above considerations is that a single-step solution is probably not attainable; we must perform a sequence of grouping, filtering, and information-reduction steps to eliminate unlikely candidates as early in the process as possible, and then make our final selection on a greatly simplified reduction of the originally presented data.

4.1 A Combinatorics Based Architecture for LD

In general, we must address two sub-problems: (1) selecting/partitioning the actual path-points from the set of potential path-points, and (2) sequencing the selected path-points.

Let us assume that we are given an unordered col-

lection of points that actually constitute the solution (rs). A very reasonable ranking function, based on the primary Gestalt property of proximity, is linear density, defined as the number of path-points per unit path length. This criteria selects the shortest path that contains all the given points. What we have just established is that a simplification (sub-problem) of our original problem is the Traveling Salesman Problem (TSP) if the solution is closed, or the problem of finding a "messenger" (open) path. Both the TSP and the messenger path problem are known to be computationally intractable for large values of n (NP-complete). For example, (at least) until recently, the largest value of n for the solution to a non-contrived TSP was 318 cities.

It is clear that in order to solve the unconstrained delineation problem we must strictly limit the the number of points that can be arbitrarily sequenced, or we must limit the number of choices that are the possible successors of any given point, or use some combination of the two preceding constraints. We have observed that we can usually find very dense path segments (greater than some minimal length related to visual detection criterion), and place perceptual and/or application-domain-related constraints on linking possibilities for these dense segments. To the extent that most of the path-points are already sequenced as members of the detected segments, and it is only the segments that must be sequenced, and even here there are only a few linking alternatives for each of the segments, we can solve the rs problem even though it is formally intractable.

Our overall approach, as shown in Figure 1 then is:

1. Detect potential road points in black/white panchromatic images and construct a binary representation that retains the perceptually obvious linear structure. This information reduction step is critical in allowing us to employ very efficient and expressive graph-theoretic methods to solve the delineation problem (see 2 and 3).

2. Assemble potential path-points into dense segments by using a fast Mimimum Spanning Tree (MST) algorithm (although the MST does not actually assure the densest connectivity, it usually provides a very good approximation to this condition). Recover the longest smooth segments – consistent with generic perceptual connectivity criterion – that can be extracted from the forest of trees generated in this step. The input to this step is the binarized image; the output is a list of disjoint (potential road) segments called RPATHS (see Figure 4).

   In a typical road delineation problem we started with a 768X638 pixel image (489,984 points) and constructed a binarized version with 55,480 potential road points. As a result of step 2, we extracted 340 segments (RPATHS) containing 21,255 points.

3. Repartition and semantically filter the collection of RPATHS to eliminate perceptual and semantic linking mistakes and irrelevant paths introduced or retained by the limited flexibility of the MST algorithm/representation and the generic parsing process.

4. Use a recently developed linking technique and representation schema, capable of expressing arbitrary perceptual and semantic constraints, to imply a network of paths that is very likely to include the road-network to be modeled. (see Figure 5).

5. Refine the 3-D geometric shape and location of the road-bed.

6. Employ contextual and semantic knowledge to produce a final enhancement of the fully automatic phase of road-model construction. Eliminate obvious errors and flag dubious entries for interactive editing.

Finding roads is decomposed into the problems listed above because, as noted earlier, the more direct approach of selecting and evaluating arbitrary paths is computationally infeasible: the combinatorics of indiscriminant search makes it simpler to first solve the more general problem of finding (generic) linear structures, and then filter out roads, than to examine all possible paths in an image to find the roads directly.

The detection task must address the problem that it is often impossible to distinguish roads from other

Figure 4: Disjoint (potential road) segments, called *RPATHS*.

Figure 5: Final output from the low-resolution road-extraction phase.

natural or man-made structures at a local level. For example, if we look at an image through a small peephole, objects such as a section of a river, or a parking lot, or the roof of a rectangular building, can all have an appearance similar to that of a small stretch of road. It is also the case that tunnels, trees, buildings, clouds, and so forth, can occlude of some section of a road causing an apparent break in road continuity.

The generic linking must address a number of problems problems. Given that the detection task is not error free (i.e, we expect to have both false alarms and misses) the implied connectivity is ambiguous or possibly incorrect. There is also the computational problem of actually linking the individually detected road points into continuous segments and connected networks that represent the road structures we are searching for.

Because of possible errors in assumed connectivity, and because roads and other linear structures may be intermixed at the level of generic linking (e.g., some linked path could be a composite of a road segment attached to some other non-road object), subsequent steps in the road delineation process must permit some relinking as well as the recognition of road versus non-road segments in the networks returned by the generic linker.

The major components of the low-resolution portion of the LD system are described in greater detail below:

4.2 Detection, Binarization, and Generic Path Formation

We have examined a number of distinct approaches to automating the delineation problem including (a) Dynamic Programming [Fischler *et al.*, 1981, Iverson, 1997] which is capable of finding a least-cost path in a real-valued 2-D array (which could be the original picture, or some derived overlay called a *cost image*), and (b) a number of graph-theoretic techniques which, in practice, require an early binarization of the input image [Fischler and Wolf, 1983, Fischler, 1994, Fischler, 1997].

Dynamic Programming (or any other global optimization technique) that can operate on the actual input data becomes computationally infeasible for anything other than cost/objective functions that are very local in nature, i.e, the cost of a path going through a particular pixel in an image should only be a function of an attribute list attached to that pixel and (say) the cost of appending the given pixel to a path that passes through an adjacent pixel–rather than being dependent on, for example, the specific positioning of the previous five pixels in the curve segment to which attachment is being considered. Thus, the nominal generality of full global optimization is not really attainable because of computational considerations. Even if we could contend with the computational difficulties, there is the further problem of actually specifying the global cost/objective function that approximately models our perceptual behavior in graylevel images. This is an even more difficult unsolved problem.

We have found, through a combination of theory and experiment [Fischler and Wolf, 1983, Fischler, 1994, Fischler, 1997], that it is possible to automatically construct a binary overlay, of almost any non-contrived graylevel image, that will retain the perceptual saliency of the linear structures (paths). It is further the case that it is now (in the binary image) possible to define the primary cues that underlie our perception of a line or path: relative proximity and smoothness of the binary (1 or 0) pixels defining the line/path. Although not a traditional Gestalt property, persistence (e.g., coherent path length) is also a cue of major importance; the other Gestalt cues play a role only when there is ambiguity due to contending interpretations, or when we recognize some known shape or repeated structure.

Generic (perceptual rather than application dependent) clustering and linking are effectively (but not perfectly) achieved by employing a modified MST algorithm with a bound on interpoint distance. The MST algorithm we devised for this purpose can be made to run in time proportional to the number of points being processed (because the points are represented by bounded integer coordinates, their density is not arbitrary).

Thus, the result of the first processing stage in our road modeling system is a collection of disjoint MST's which can be separately parsed to to provide a collection of line-segments (RPATHS) as the final output of the generic linking step .

This parsing process involves (1) finding a primary path through the tree (typically a diameter path), (2) trimming back branches with ragged ends, (3) pruning short branches, (4) partitioning the remaining collection of branches into disjoint paths which are pair-wise linked at the MST nodes according to geometric and (original-image) intensity smoothness criterion. An example showing the result of this process is presented in Figures 2-4.

4.3 The Semantic Filter (SF)

The purpose of the semantic filter is to extract, from a collection of perceptually salient paths, those sub-paths that are compatible with the constraints of some specified application or purpose (in this case, sub-paths that could be road segments in an aerial image).

This system component takes as its input a list of generic perceptually-salient paths (RPATHS) and produces, as its output, a list of path-segments (RPATHS-F). Each item (called a seg) in RPATHS-F, is a coherent sub-path of some path in RPATHS; the segs returned in RPATHS-F are open and non-self-intersecting, and any pair of segs are disjoint with the possible exception of a single intersection-point (as are the paths in RPATHS).

The SF processes each path in RPATHS independently. It first partitions the path at it's salient points using the algorithm described in [Fischler and Wolf, 1994]. This partitioning step is necessary to recover components of the application relevant paths that were combined with other (incidental) adjacent paths in the original image. Each seg produced by the partitioning process is evaluated for compatibility with the constraints of the intended application on an accept or reject basis. The accepted segs are appended to the output-list *RPATHS-F*.

While in theory, the semantic filter might have to be completely redesigned for each new application, we have found that the same set attributes (properly parameterized for the different applications) appears adequate for such diverse tasks as finding roads or rivers in aerial images, and for finding man-made objects (e.g., building edges) or natural objects (e.g., the skyline, tree-trunks) in ground-level images.

The attributes we currently evaluate are concerned with length/coherence, directionality/purposiveness, smoothness, and degree-of-randomness:

1. Length. Very short segs are typically rejected as being clutter or unimportant (they can be recovered later if needed); very long segments are typically accepted since they are too important to discard without the further analysis to be performed later.

2. Directional consistency. Consistency of global direction based on a histogram of the directions between adjacent seg pixels obtained from a chain-coded representation of the seg.

3. Smoothness. This property is measured in two ways. First, each seg is inherently smooth to some degree because its parent in RPATHS was partitioned into segments at salient (or high curvature) points. Thus, the length of the seg is an indirect measure of its smoothness (the longer the seg, the smoother it is). Second, we measure the seg's deviation from a best fitting circular-arc to look for a smoothness property that is especially important for some applications (e.g., finding man-made objects including roads and streets).

4. Randomness. We have devised a weak measure of repeated structure (e.g., symmetry), in a path; this measure together with the evaluation of coherent length, consistent direction, and smoothness, provide a basis for judging whether a seg is a "purposeful" or an apparently random structure.

In the example shown in this paper (Figures 2-4) there were 146 RPATHS containing a total of 9517 pixels. The semantic Road Filter extracted 65 segs containing a total of 4427 pixels from the given RPATHS.

5 The Semantic Linker (SL)

The purpose of the semantic linker is to combine all the segs in the list RPATHS-F (produced by the Semantic Filter) into a network of unpartitioned

paths. If it were the case that all the components of a graph (in the mathematical sense) were present, the design of an efficient linker would still pose some significant software problems, but conceptually, would be straight forward. In actuality, the segments (graph edges) we must assemble into the "road-graph" frequently have gaps and don't necessarily extend to their true point of intersection with other segments. The linker must make "informed" decisions as to how to complete the graph. (There are other problems to address, including the aspects of a road-network not covered by graph theory, e.g., a real-world road intersection can be a significantly sized area rather than a single point; or, there can be more than one road linking two intersections; or, a single edge can have both its endpoints located at the same intersection. There are also the problems associated with 3-D intersections such as highway overpasses.)

The input to SL is RPATHS-F, and its output is a table defining the road-graph that represents the topology of the road network. The SL (conceptually) examines every pair of segs in RPATHS-F and determines if they intersect at some point interior to both segs, or if a small extension of one seg will intersect the other, or if they can be adjacent components of an extended path compatible with the constraints of the specified application. After all the link decisions have been made, a clustering operation is used to group the locations, at which the links between segs have been established, into the vertices of the graph.

The SL typically uses three types of criteria to make a link decision for a pair of segs:

1. The relative geometric positioning and separation of the segs. For example, in the case of road delineation, the criterion is typically a bound on the separation-distance between nominally corresponding endpoints (one on each seg).

2. Global attributes of the segs. For example, we can require that the spectral distribution, or image intensity, or mean width of the two candidate segs be identical to within some specified tolerance.

3. Acceptance by the semantic filter. If the two candidate segs are sequentially linked and treated as a single seg by the semantic filter, is the combination accepted or rejected.

5.1 The High-Resolution Analysis Phase

As currently implemented, the result from the semantic linker phase is a 2-D network of 8-connected pixel chains that correspond to the road centerlines in a single, low-resolution, image of the study area. The high-resolution 3-D phase uses this result to "index into" a collection of overlapping images. In order to do this, the pixels chains are projected into object space, by monoplotting the pixel coordinates against a terrain elevation model and collecting them into object-space curves. To reduce the redundancy in the data and remove the artifacts due to the integer calculations of the previous stages, the curves are resampled, snaked, and generalized to derive a piecewise linear, real-valued, curve that closely approximates the centerline of the roads in the network.

We then consider the entire collection of high-resolution images available and for each segment of each road and build an initial road segment visibility table that indicates in what images a given segment should be visible disregarding interobject occlusions. We then make a second pass through the table and check if any other already modeled objects in the scene obscure the segments. This mechanism is implemented in a very general way, so that any objects already modeled are checked.

In the case of recent demonstrations using the Ft. Benning dataset, we are essentially in a cold-start mode and the only apriori spatial objects we have are the two terrain elevation models: NIMA DTED2 that characterizes the topography of the bare earth and ERIM IFSARE that captures the shape of the top of the vegetation. This allows us to check for and remove images from the entries in the road segment visibility table, where that segment is not visible due to occlusion by the tree canopy.

Finally, we filter the table based on sensor type, acquisition geometry, and local contrast and resolution of the image where the segment appears.

At this point, we invoke the SRI Model-Based Optimization system on each of the road segments

in the network, using all of the images that have passed the above tests.

This system has been described in detail in other papers [Heller *et al.*, 1996, Fua, 1996], but briefly, it attempts to align the two edges of the road with high gradients in the images while not deviating significanly from an apriori geometric model of a generic rural road (e.g., slowly varying width, no sharp bends). When operating with multiple images, as is the case here, the 3-D road model is projected into each image, the line integral of the gradients and its partials with respect to the ribbon parameters is computed; this information is propagated back to the 3-D ribbon, the ribbon is adjusted and then reprojected into the images. This iterative process is repeated until no change in the ribbon's parameters are made to a maximum of five iterations.

The optimization take place in two phases, first overall width of each the road in the network is adjusted in 3 meter steps from 3 to 24 meters. The width that provides the maximized the line integral of the gradient is retained. Then the full optimization is run, which adjusts the 3-D position, surface normal, and width of the road at each point along the road. To prevent corruption of the connectivity of the network, the junctions ("nodes") in the road network are constrained to their initial locations and not adjusted during the optimization. The full result of this phase is shown in Figure 6.

Experiments have shown that using more than four or five images for each road does not provide significant improvement and in some cases can degrades performance. Therefore it is important be discriminating and choose a few good images rather than a large number of poorer ones. It is this observation that motivated the development of the extensive filtering described above.

6 Evaluation Rationale and Metrics

We assume that the computer cost and time required to run a typical feature extraction algorithm on an image will continue to decrease, and will be insignificant within a five to ten year time frame. Thus, in a practical setting, assuming that the computer time needed is not excessive (i.e., many times longer that doing the job manually), the cost of automation largely amounts to the time spent fixing the errors and short-commings of the automated process.

Our primary practical concern is in reducing human interaction time. Hence our primary benchmark evaluation metric focuses on this quantity, human interaction time. Nevertheless, from both a scientific and long-range practical perspective, measuring progress toward a fully-automated feature extraction capability is also an item of major concern and we separately evaluate the performance (with respect to running time, completeness, and correctness) of the fully-automated component of our system.

The *goal* of the evaluation we present later in this document is to quantify the performance (in terms of human-interaction-time, correctness, and completeness), of our overall system and its automated component, to recover a road-centerline data-model for the Ft Benning McKenna MOUT facility and surrounding area.

A Reference *Road-Centerline Data-Model* includes the specification of a collection of Reference-Road-Segments (RRS). Each such road-segment is an ordered list of geo-referenced 3-D coordinates representing a sampling of points along the centerline of a road-segment; the road-segment-centerline is assumed to be a continuous path in 3-D space, and the gaps between sample points are straight-line (in a cartesian system) extents of the segment-centerline.

Given that there are road-like entities in an image that are ambiguous over some portion (or all) of their extent, with respect to their classification (e.g., a long dirt road that gets continuously narrower until it finally disappears; the point at which the road changes into a path is ambiguous), two different cartographers, using their best judgment, might assign different labels to (portions of) such objects. For this and other reasons, the Reference-Model (RM) can include *Don't-Evaluate* volumes, regions, or segments. In essence, these are portions of the scene which are excluded from the evaluation.

The Road-Centerline Data-Model includes a list of nodes (3-D spatial locations) denoting the locations at which the RRS intersect. This information

Figure 6: Example result from the High Resolution Analysis Phase.

describes the Reference-Road-Graph topology.

The Road-Centerline Data-Model includes a collection of attributes for each RRS, including number-of-traffic-lanes, minimum-useable-width, surface-material-type, etc.

6.1 The Evaluation Process

The evaluation metrics (correctness and completeness) are based upon the following definitions and tabulated quantities:

Reference Model An object space model generally recognized as representing the "correct" answer for the feature extraction task under evaluation.

Derived Model An object space model created by the algorithm or system under evaluation.

True Positives (TP) Length of road that, within a specified tolerance, is common to both the Derived and Reference Models.

False Positives (FP) Length of road that appears in the Derived Model but not in the Reference – even when we dialate the reference to include all derived road-segments within some tolerance area or volume around the Reference segment.

False Negatives (FN) Length of road that appears in Reference but not in the Derived Model – even when we dialate the Derived Model to include all Reference-segments within some tolerance area or volume around the Derived segment.

Explicit algorithmic definitions of the above quantities are provided in a separate paper in this volume [Fischler and Bolles, 1998].

From these tabulated quantities, the following metrics are calculated.

Completeness: The percentage of a specified class of objects included in the reference model that also appear in the derived model. This metric corresponds to what has also been called "detection percentage:"

$$100 \times \frac{TP}{(TP+FN)}. \qquad (1)$$

It has a range from 0-100% (a large value is good).

Correctness: The percentage of some specified class of objects included in the Derived model that are also included in the Reference model.

$$100 \times \frac{TP}{(TP+FP)}. \qquad (2)$$

It has a range from 0-100% (a large value is good).

7 System Performance

The formal Year1 Road Modeling demonstration was presented at SRI on 22 April 1998.

The reference benchmark timing result, a road model (with some minor errors) of the Ft. Benning MOUT site produced by a professional cartographer was 279 minutes. Our goal in this Year1 demonstration was to use our computer system to model the same site in an order of magnitude less time for the needed human interaction in a final editing step (computer time was recorded as part of the experimental record, but did not enter into the evaluation).

The automatic road extraction process (the low-resolution process followed by the 3-D multi-image refinement process) took approximately 10 minutes of computer time on an SGI R10000 O2. (We later ran the same analysis on a newer machine (a SUN Ultra30) and it only took 5 minutes.)

The evaluation of the automatic road extraction results were as follows:

Correctness: 90% Completeness: 86%

After interactive editing, these scores increased to:

Correctness: 98% Completeness: 93%

We have run the editing process 3 times for timing purposes. The human interaction time varied from 22 to 25 minutes, always under our goal of 27.9 minutes.

8 Current Status and Future Plans

Our effort to design a baseline linear delineation (LD) system as part of the BOS architecture, and to integrate it into to the RCDE system for evaluation and testing is complete [Fischler *et al.*, 1998]. In addition to the interactive "Snake" and correlation-tracking algorithms already resident within the RCDE, we have selected components from the SRI low-resolution generic-LD-research-system and reimplemented the code as needed for RCDE compatibility.

We also made a number of modifications to the LD code to enable processing of the large images typically encountered in cartography and intelligence applications. Our current implementation can demonstrate a significant advance over previous state-of-the-art performance in fully automated road modeling. Current on-going work involves putting the road modeling system under complete CBACS supervision to exploit contextual information, adding additional components to deal with urban streets, and extending the core algorithm to exploit the information available across a range of scales of resolution.

We believe we have developed one of the best and most complete collections of algorithms for linear delineation available anywhere in the world. Nevertheless, to achieve the ultimate goal of completely automated robust delineation of roads (streets, etc.) appearing in aerial images, we must solve some additional problems.

Self-evaluation. The most important problem we face in assembling a fully automated road delineation system, which has little or no need for final human editing, is to eliminate errors (either of omission or commission) that would lead a naive user of the product to question its credibility. The system cannot afford to miss a road that any human observer can easily detect, or to insert a road that obviously isn't present. This means that the system must be capable of a high degree of self-evaluation – it must be able to access and employ enough context to be very sure of the answers it produces (at least) in obvious situations.

Operation in complex scenes. In complex environments, such as in urban scenes, streets, buildings, and trees form a minimal contextual unit. A delineation system for streets has no hope of obtaining reasonable performance unless it also "understands" buildings and trees, and how they "interact" with streets. The idea of a simple stand-alone algorithm to perform a complex recognition task is not viable. Our current approach to structuring the APGD task within the framework of a context-based architecture is still in an early state of development, but the ultimate success of our efforts will depend on easy and effective access (by the feature extraction algorithms) to high-level contextual and semantic knowledge.

A Assumptions for Simple/Rural Road Delineation

The imaged roads are perceptually detectable in an available (possibly reduced resolution) single synoptic view, with no special effort or detailed study, as smoothly curving line-like objects with no internal detail. The roads are purposive as transportation-links; they have direction consistency, tend to follow elevation contours and have switch-backs only in steep terrain. Rural roads lie on the earth's surface, have few major intersections (or small closed circuits), and are uncorrelated with other nearby man-made structures, including other roads.

Acceptable resolution: The image, or a reduced resolution version of the image, depicts the roads with an width of two to five pixels.

Good contrast: The roads appear dark or light against the background.

Good visibility: The roads are mostly unoccluded.

References

[Bolles *et al.*, 1979] R.C. Bolles, L.H. Quam, M.A. Fischler, and H.C. Wolf. Automatic determination of image to database correspondence. In *IJCAI79*, pages 73–78, 1979.

[Fischler and Bolles, 1998] Martin A. Fischler and Robert C. Bolles. Evaluation of a Road-

Centerline Data Model. In *DARPA Image Understanding Workshop*, 1998.

[Fischler and Wolf, 1983] M.A. Fischler and H.C. Wolf. Linear Delineation. In *Conference on Computer Vision and Pattern Recognition*, pages 351–356, June 1983.

[Fischler and Wolf, 1994] M.A. Fischler and H.C. Wolf. Locating perceptually salient points on planar curves. *PAMI*, 16(2):113–129, February 1994.

[Fischler *et al.*, 1981] M.A. Fischler, J.M. Tenenbaum, and Wolf H.C. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. *CGIP*, 15(3):201–223, March 1981.

[Fischler *et al.*, 1998] Martin A. Fischler, Robert C. Bolles, Aaron J. Heller, and Christopher I. Connolly. An Integrated Feasibility Demonstration for Automatic Population of Geospatial Databases. In *DARPA Image Understanding Workshop*, 1998.

[Fischler, 1994] M.A Fischler. The Perception of Linear Structure: A Generic Linker. In *DARPA Image Understanding Workshop*, Monterey, CA, November 1994.

[Fischler, 1997] M.A. Fischler. Finding the perceptually obvious path. In *DARPA97*, pages 957–970, 1997.

[Fua and Leclerc, 1990] P. Fua and Y. G. Leclerc. Model Driven Edge Detection. *Machine Vision and Applications*, 3:45–56, 1990.

[Fua, 1996] P. Fua. Model-Based Optimization: Accurate and Consistent Site Modeling. In *XVIII ISPRS Congress*, Vienna, Austria, July 1996.

[Goddard, 1996] Greg Goddard. Ft. Benning GA McKenna MOUT, Database Generation. Final report, GDE Systems, Inc., San Diego, CA, March 1996. Available from `http://www.ai.sri.com/~apgd/vl/datasets/Benning/db-report/`.

[Heller and Quam, 1997] Aaron J. Heller and Lynn H. Quam. The RADIUS Common Development Environment. In Oscar Firschein and Tom Strat, editors, *RADIUS: Image Understanding for Imagery Intelligence*. Morgan Kaufmann, San Mateo (CA), 1997.

[Heller *et al.*, 1996] A. J. Heller, P. Fua, C. Connolly, and J. Sargent. The Site-Model Construction Component of the RADIUS Testbed System. In *DARPA Image Understanding Workshop*, pages 345–355, 1996.

[Iverson, 1997] L. Iverson. Dynamic programming delineation. In *DARPA97*, pages 951–956, 1997.

[McKeown and Denlinger, 1988] D.M. McKeown and J.L. Denlinger. Cooperative methods for road tracking in aerial imagery. In *CVPR88*, pages 662–672, 1988.

[Nevatia and Babu, 1978] R. Nevatia and K.R. Babu. Linear feature extraction. In *DARPA78*, pages 73–78, 1978.

[Quam, 1978] L.H. Quam. Road Tracking and Anomaly Detection. In *DARPA Image Understanding Workshop*, pages 51–55, May 1978.

APPENDIX 2:

Automated Road Modeling: Constructing the Road Graph

Automated Road Modeling: Constructing the Road Graph

Martin A. Fischler

March 5, 1999

1 Introduction

This document describes an algorithm that derives an explicit 2-D graph from a collection of line (road) segments. For the purposes of *road modeling*, the line segments are assumed to be the center-lines of road-segments detected in aerial imagery with a ground surface resolution in the range of 2-10 meters per pixel – i.e., a resolution range in which the internal road structure is not visible and the roads appear as thin lines; if necessary, the original image resolution is reduced to obtain this condition. The data model for the derived graph makes explicit both the geometry and topology of the road-network being modeled.

In [Fischler and Heller, 1998], we described a complete system for modeling the 3-D geometry and various physical attributes of the network of roads appearing in a collection aerial images of some specified geographic area. The system block-diagram (Figure 1) shows a sequence of steps leading to the block entitled *Network Linker* – actually the semantic network linker SNL; this is the component of the system we are primarily concerned with here. It will be assumed that, in general, there will be some errors in the list of segments (called *road-segs*) provided as input to the SNL. In particular, because of occlusions or lack of adequate contrast, the segments provided as input may be shortened (e.g., do not extend to their actual intersection points with other segments); may be incorrectly partitioned because of the loss of an interior subsegment; or may be missing altogether. Thus, the SNL must make "informed" decisions as to how to complete the graph in the absence of complete information.

There may be non-road structures (e.g., a river) in *road-segs* that managed to pass through the earlier filtering processes, and we must also address those aspects of an actual road-network that are not covered by graph theory, e.g., a real-world road intersection can be a significantly sized area rather than a single point; or, there can be more than one road segment linking two intersections; or, a single segment can have both its endpoints located at the same intersection. Because of these and other problems, the SNL must be able to exploit context and semantic knowledge relevant to road construction/formation,[1] rather than just making explicit the generic connectivity relationships inherent in the given data (independent of the problem domain).

2 Algorithm Architecture

A basic function of the Semantic Network Linker is to describe the explicit (literal) topology of the graph formed by the individual segs in the list *road-segs* (produced by the Semantic Filter). In this most primitive version of the linking operation, the SNL performs a purely passive and syntactic (problem-domain-independent) operation – it constructs the

[1] Existing terrain features can be used as roads; e.g., a dry river-bed or wash

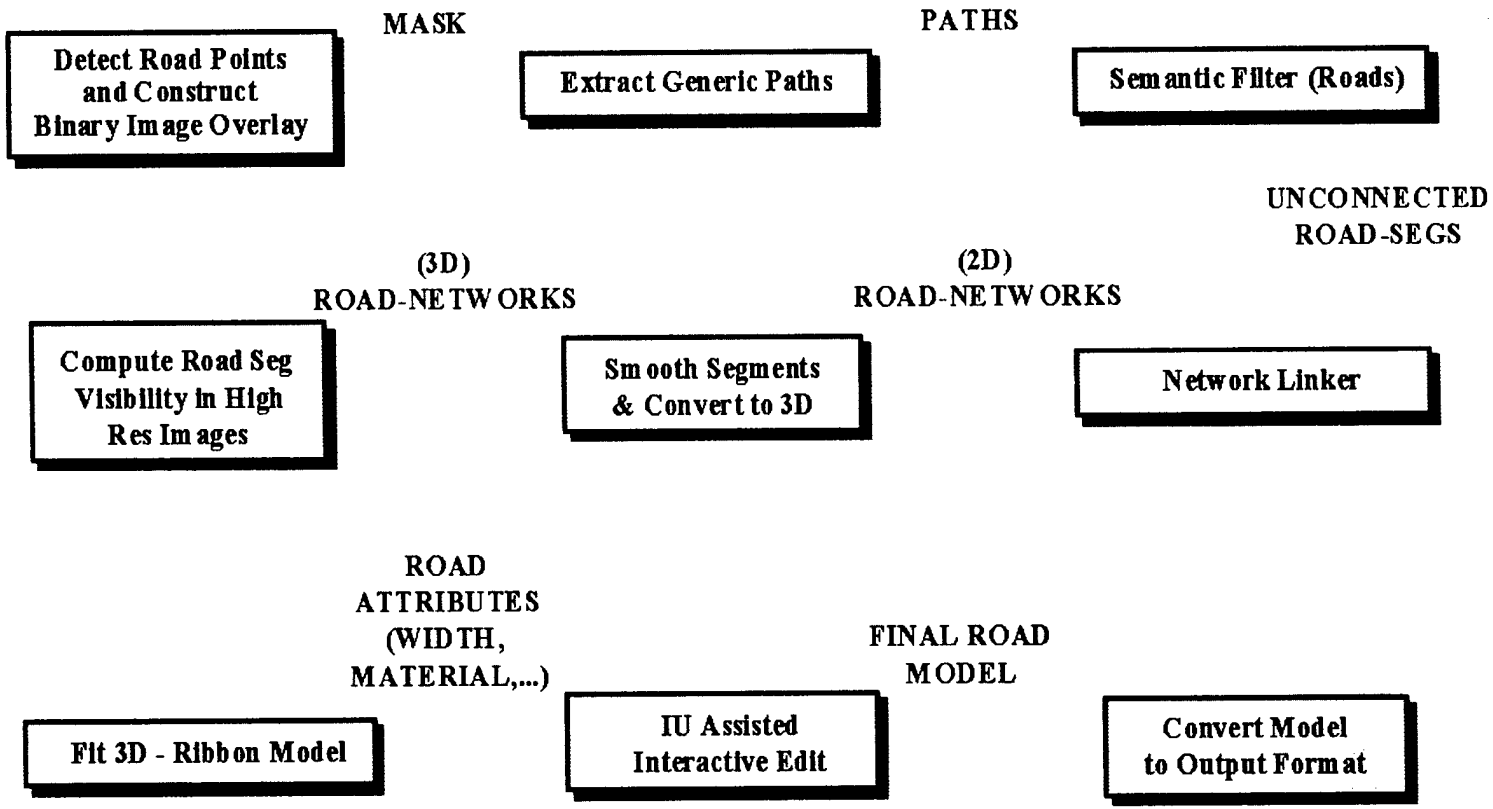


Figure 1: Block diagram of the end-to-end Road Modeling System.

data-structure called *vertex-table* that identifies the points of intersection of the members of *road-segs*; and also catalogues in *vertex-table* (as degree-1 vertices) the unlinked endpoints of the members of *road-segs*.

In actuality, because *road-segs* almost always contains omission errors due to occlusions and areas of very low contrast, the most critical function of the SNL is to deduce the presence of vertices (road intersections) implied by, but not explicitly present, in the basic syntactic analysis. This deductive process is based on both generic perceptual cues and also on the semantics of road-network construction/formation.

We note that at a resolution where roads are lines with no obvious width, vertices (intersection points) have no independent visible presence, they are an artifact of the intersection of two (or more) roads. Since vertices are ephemeral in the above sense, they are always computed from a collection of road-segs called *road-segs2*; the SNL constructs *road-segs2* from *road-segs* by adding new segs to this basic collection. These new segs are the deduced completions of the initial (input) road-graph and are typically confirmed by visible evidence in a directed search of the original image or the derived MASK.

The SNL performs the following sequence of operations:

1. The SNL considers all concatenated pairs of segments that can be formed from the entries in *road-segs* and determines if each such resulting path satisfies a collection of perceptual and semantic conditions on line-appearance and road formation. All such accepted pairs are listed in a data structure called *link-pairs*

2. The connectivity information provided by *link-pairs* allows the SNL to perform a directed search for additional line segments whose presence would result in a simpler road-graph (e.g., fewer vertices). These additional segments, that fill "gaps" in the unaugmented graph implied by *road-segs*, are currently found by a more tolerant localized search in the MASK that produced *road-segs*, but they could also be provided by other means (e.g., going back to the intensity information in the original image, or even a second image of the same site). This "filling-in" operation is very conservative; if there is more than a single obvious choice, no action is taken. The result of the directed search is an augmented set of road-segments called *road-segs2* and a correspondingly more complete set of *link-pairs*. A set of syntactic operations performed on these data structures results in the formation of the data structure called *vertex-table* which defines (the current version of) the desired Road-Graph.

3. Given the additional context provided by the first compilation of a road-graph for the given site, we can now apply additional semantic rules that are applicable (only) when a road-graph is available. E.G., in a "large-enough" synaptic view of the site being modeled, (by definition[2]) all roads form a single connected network. Thus, small isolated components of the initial graph can be deleted. This, in turn, reduces the combinatorics for a more extensive

[2]Section 6 contains definitions of important terms and symbols.

search for missed connections in the retained (valid) portion of the road-graph. In particular, we perform a focused search for extensions of road-segments that terminate in degree-1 vertices. Now, rather than requiring positive evidence for a reasonable extension to fill a gap, we can use lack of negative evidence to take this action. The net result of the above strategy is an iterative boot-strapping enhancement of the road-graph by 2 or 3 repeated applications of the (same) SNL algorithm – in each repetition, the road-graph becomes simpler and more completely connected.

3 Algorithm Description

The input to SNL is *road-segs*, and its output is *road-segs2* and the table *vertex-table* defining the road-graph that represents the topology of the road network. The SNL (conceptually) examines every pair of segs in *road-segs* and determines if they intersect at some point common to both segs, or if a small extension of one seg will intersect the other, or if they can be sequential components of an extended path compatible with the constraints of the specified application. After all the link decisions have been made, a clustering operation is used to group the locations, at which the links between segs have been established, into the vertices of the graph.

The SNL uses three types of criteria to make a link decision for a pair of segs (corresponding to the *Gestalt* criterion of proximity and good-continuation):

1. *The relative geometric positioning and separation of the segs.* This criterion is expressed as a bound on the join-angle and on the separation-distance between nominally corresponding endpoints (one on each seg). This test provides a necessary but not sufficient condition for the presence of a new link-segment unless the separation distance is less than the tolerance for the positioning of an intersection (5 pixels or 25 meters). In the later case it is not necessary to add a new segment to *road-segs2*, the intersection will be noted by the SNL and a corresponding vertex will be added to *vertex-table*.

2. *The actual presence of a visible link that was not captured or retained in *road-segs*.* Given two segment-endpoints in close proximity, we identify a window in MASK containing both these vertices and using a single-path finding algorithm (the algorithm "F*" described in [Fischler *et al.*, 1981]) determine if an acceptably dense path actually exists between the given vertices – if so, it is appended to the list *road-segs2*.

3. *Acceptance by the Semantic Filter* (figure 1). The two candidate segs are sequentially linked (the gap between their endpoints filled by a straight-line[3]) and treated as a single seg by the semantic filter; if the combination is accepted, the straight-line-connector is appended to the list *road-segs2*.

3.1 Directed Search for Road-Graph Completion

In attempting to produce a complete road-graph, beyond the explicit data provided

[3] A spline could be used, but for small gaps, the extra complexity is not justified

in *road-segs*, the SNL must make some "informed" decisions (based on context and semantic knowledge) about how large a gap in the network can be filled without direct evidence; and from a computational standpoint, there is the problem of how big an area to search for a potential continuation of a terminated road-segment. Because of the way *road-segs* was created, and because of the combinatorics involved, all extensions to *road-segs* are required to include the end-point of at least one segment in *road-segs*. Given such an endpoint, the SNL performs four types of search for a continuation:

- An area search (for another segment endpoint) centered on the given endpoint, with a diameter between 50-150m (10-30 pixels). The lower-limit of the search dimensions is based, in-part, on the definition of a *road* given in section 6; the upper limit is based on the combinatorics involved. A link-segment is added to *road-segs* if there is evidence (i.e., a sufficiently dense path) in MASK that a road is actually present. However, if the two endpoints involved are degree-1 vertices of the current road-graph, and the associated segments pass the test for "good-continuation," a link-segment joining them is added to *road-segs* without further evidence.

- An area search for any portion of another segment within 5-10m of the given endpoint based on the definition of a road given in section 6. If found, a link-segment is added to *road-segs* without any further verification required.

- If the given endpoint is a degree-1 vertex of the current road-graph, a 100m-long search along a straight line extrapolation of the tangent to the corresponding segment is made for an intersection with any portion of another segment. If *exactly* one such intersection is found (called a "T-intersection"), a link-segment is added to *road-segs* without further verification. We note that if the length of the search path is increased, there is a higher probability of finding more than one intersection along the path, and thus, we might actually lower rather than increase the probability of adding a new link-segment to the road-graph.

- If the given endpoint is a degree-1 vertex of the current road-graph, a focused search for another vertex of the road-graph is conducted in a narrow rectangular region whose axis is the straight line extrapolation of the tangent to the given segment. The length of this search is set by the local context, but is generally in the range of 30-60 pixels. A link-segment joining the two endpoints, is added to *road-segs* if there is a sufficiently dense path between these endpoints (in MASK) to verify that a road is actually present. If more than one vertex is found in the search-area, only the closest one is considered. The use of this search procedure is the main feature that distinguishes the "augmented" from the basic-SNL.

3.2 Implementation

The complete algorithm depicted in the Block-Diagram (figure 1) has been implemented in Common-Lisp and runs on a low-end Silicon-Graphics or Sun worksta-

tion. The 665 x 365 ALV-image,[4] discussed in a following section, required less than three minutes of computer processing for the system to produce the completely linked low-resolution road model (output by the SNL). The SNL run-time was one minute.

A key implementation requirement is that the modeling system (except during development) should be able to run on images of arbitrary size – we have run successful experiments on images as large as 20,000 x 20,000 pixels. To achieve this goal, we restricted the algorithms to (nominally) linear, but no worse than *nlogn* computational complexity. Images are stored and processed as partitioned blocks. All interactions are assumed to occur between entities that have some known maximum spatial separation, and thus, higher-level derived quantities can also be stored in cells such that only a few cells ever have to be considered for any given computation.

4 Algorithm Performance

The SNL's main responsibility is to recover the underlying topology of the road-graph implied by the list of road segments (*road-segs*) provided as input. Side effects of the operations performed by the SNL include "filling-in" of some of the small gaps in the input, and elimination of (nominally) non-road-segments detected by their isolation from the main body of the final road-graph.

The algorithm is expected to correctly recover the connectivity relations explicit in the input, and make human-level judgments for the implicit connections – given that such a representative human has no additional sources of information beyond the given input (*road-segs*). We also assume (for the current version of the algorithm) that we are dealing with rural scenes in which the roads are relatively uncorrelated with each other, have relatively few intersections, and occupy less than (nominally) 1/50th of the image.

Figures (2) and (3) show the syntactic performance (explicit recovery) of the SNL given a complete grid. The grid is a difficult test case for the SNL which currently assumes a rural setting. The given grid has a line and intersection density of approximately 10 times that expected; never the less, the recovered geometry and topology are correct. The algorithm properly identified the actual intersections and did not introduce any incorrect links. In figures (4) through (6) we observe the conservative behavior of the rural-SNL in a complex environment – it did not introduce any incorrect links, but a number of links that were very reasonable extrapolations of the given data were omitted. In figure (7), we augmented the SNL with an experimental module[5] being designed for urban environments – here the results are more complete. A correct answer (i.e., that produced by most people given the image and the information that the grid is an actual street-pattern and is not necessarily complete) would probably be very similar to that produced by the augmented-SNL. Figures (8) through (11) show the performance of the algorithm on an aerial image.

Neither the basic nor the augmented SNL has any explicit knowledge of straight lines or grids at present, but such knowledge is valuable in modeling urban streets and it will be added to the augmented

[4] alv-2-44-full.gl UTM-window: ((413 2599) (1743 3329)); 2m per pixel

[5] The command sequence used was (get-road-model test-image1) (citi-link *road-segs*)

SNL in the near future. The basic SNL is the unaltered module, with no parameter changes, that has been used in all our benchmark tests and related demonstrations [Fischler and Heller, 1998].

Figures (12) through (20) show a rural mountainous environment (with some industrial roads[6] and buildings) that is more representative of the type of contextual setting the algorithm expects. The performance of the SNL-algorithm[7] here is very close to what a human would produce over most of the scene (even though there are some minor errors in the upper-left corner) – especially if the human did not see the original image (figure 12), but only the input[8] provided to the SNL (figure 18).

5 Discussion

A key problem we face in trying to duplicate human performance in deriving a road model from imagery is having some way to determine if a proposed model (or model-component) is "correct." At present, this problem is unsolved without recourse to either human intervention, or to information beyond that directly available from the imagery. On the other hand, there are many fairly reliable ways to determine if a small road-model component (especially, an isolated road segment) is likely to be incorrect. Our approach, therefore, has been to generate a collection of small feasible components, filter-out the *unlikely* candidates, and then link the survivors into larger connected components. The main characteristic we exploit in this process is that of road-network coherence. If we can find a model whose component parts are independently likely to be correct, and whose connectivity satisfies a set of criterion that is also consistent with road-model construction/formation constraints, then the larger the size of a candidate road-model component, the more likely it is to be correct.

We identify the following essential processes:

- **Detection** of locations in an image where there is evidence of linear structure. A very effective algorithm for this purpose (described in [Fischler and Wolf, 1987]) is the basis for our construction of the the binary image overlay MASK.
- **Sequencing** of line-point locations (in MASK) into line-segments. We define sequencing as a linking operation in which there is only one reasonable simple 2-D curve that contains the given points. This is accomplished by parsing a Minimum Spanning Tree into a collection of line segments called *rpaths* (Reference [Fischler and Wolf, 1987]).
- **Filtering**. Elimination of line segments in *rpaths* that do not satisfy the geometrical (and possibly other) attributes of a road segment. The collection of segments that survive this process are called *road-segs*.
- **Linking**. The construction of a complete (road) network from the collection of segments in *road-segs*. A

[6] A few of the wide roads in the industrial area are not found at the given resolution. A second pass at a lower resolution is needed to complete the delineation.

[7] The command sequence used to produce figure (20) is (get-road-model :max-ratio .20) (extended-linkerx :long-rpaths 120) (delete-small-network-components :th 400) (delete-spurs :th 75).

[8] The basic SNL does look at the MASK when trying to confirm a hypothesized link, and the extended-linker also looks at *rpaths*

discussion of some of the underlying theory is presented in [Fischler, 1997].

We deal with the combinatorics of the linking process by only considering two segments at a time to form an initial set of "join-candidates." This initial set of candidates (*link-pairs*) implies a set of vertices of possibly high degree. The implied vertices are now examined individually, and those that do not pass a set of tests (no vertex can be more than 5-pixels/25-meters in diameter; that is, all the line segments associated with a given vertex must terminate within a circle 5-pixels in diameter; no vertex can contain both ends of the same segment unless the segment is at least 5-pixel long) are modified by either removing the links to a segment that fails the vertex membership criterion, or finding additional evidence that a segment extends to an explicit intersection with other vertex segments, or by splitting the vertex into two or more simpler ones (this operation is actually accomplished by removing links joining some of the original segments associated with the vertex).

The set of rules that are invoked to determine whether two segments (both members of *road-segs*) should be linked is based on the requirement that two segments can only be linked at their endpoints unless there is an actual interior point of intersection. We also search for "T-intersections," locations where one segment endpoint appears to intersect an interior point of a second segment. In this case, we split the second segment at the implied vertex and add the implied (short) linking segment to *road-segs2*.

A key idea in all the above is that we only make a change to the existing network in a few recognized predefined (relatively simple) situations. Complex linking situations do not have to be explicitly identified and understood by the SNL, they are resolved through an iterative sequence of more primitive simplifications. The linking algorithms are designed to produce an improved (or neutral) result no matter how often they are recursively applied. A second reason for this conservative approach is related to our strategy for integrating models (for the same site) computed at different resolutions, or over different images, or even from non-image information (e.g., existing maps).

An arbitrary aerial image of some specified site will typically not provide visual access to all the roads in the area – different viewpoints, and possibly images acquired at different times or even seasons may be required. Further, the visibility and detailed appearance of roads (as well as other features) changes as the resolution is varied. To compile a complete and accurate road model we must assume that an effective multi-source integration strategy is available. Statistical methods, normally used for multi-source integration, are not suitable for modeling the natural (unconstrained) outdoor world. However, if we can assume that our various models are reasonably close to being error-free, and we can project them into the same geographic coordinate-frame, then a superposition of models is a valid integration strategy. *This is the approach we have taken.* The requirement then is that we run our algorithms at a point of the ROC curve that produces the highest obtainable correctness at the expense of completeness – and we regain completeness by multi-source integration. We have found that even if we only have a single high-resolution image of a site, this integration strategy (using conservatively produced multi-resolution road-models) gives

much better results than any other feasible alternative tied so far.

In formal and informal testing [Fischler and Heller, 1998] on mapping quality aerial images, the SNL produced result is typically 90-100 percent correct and 80-100 percent complete.[9]

6 Definitions

Road The problems of defining a "road" in terms of its visual appearance, and of distinguishing between different classes of road-like objects, are discussed in [Fischler and Heller, 1998]. For our purposes in this document, we assume that roads form an infinite network, only part of which is present (but not always completely visible) in any given image. A completely isolated "road-like" object is not a road. A "spur," (e.g., a driveway) is considered to be a road if it is at least 300m (60 pixels) long. Two roads which parallel each-other at a separation distance of less than 50m (10 pixels) cause an ambiguous situation in which they may be merged into a single road segment. Similarly, if any pair of points, one on each of two road segments, are separated by less then 50m (10 pixels), they may be assumed to be in contact and part of an intersection.

MASK The generic line-mask; a binary overlay of the image being processed that retains the perceptual appearance of the original "line-like" structures.

EMST A Euclidean Minimum Spanning Tree. The tree is composed of sequences of 2-D points called *segments*; the intersection points of two or more segments are called *vertices*; the geometric spacing between two successive points of a segment are called *gaps*. The concatenation of two or more segments are called *paths*. The EMST is a shortest (smallest sum of Euclidean distances) tree spanning all the points in a given collection.

MASK-MST A subset of the complete Euclidean Minimum Spanning Tree that covers the points in the MASK; the subset is obtained by breaking the connectivity between points of a segment (i.e., deleting the gaps between the detected line-points in the MASK) that exceed some specified length, and pruning short/sparse paths that branch off the diameter path through the tree and the recursively formed subtrees.

rpaths A disjoint (except for intersection points) collection of paths extracted from the subtrees that comprise the MASK-MST.

road-segs A strict subset of the points/paths contained in *rpaths* that is obtained by deleting non "road-like" paths and sub-paths. Since no intersection information is retained, each entry in the list *road-segs* is formally a segment (seg) rather than a path.

road-segs2 A list of segments that comprise the branches of the derived road-graph. Nominally, a superset of *road-segs*

link-pairs A data-structure that de-

[9] **Correct** = percent of the derived road-model that agrees with a human produced reference model. **Complete** = percent of a human produced reference model included in the derived model.

fines the pairwise linking of the segments in *road-segs2*

vertex-table A data structure that defines the derived road-graph.

References

[Fischler and Heller, 1998] Martin A. Fischler and Aaron J. Heller. Automated Techniques for Road Network Modeling. In *DARPA Image Understanding Workshop*, 1998.

[Fischler and Wolf, 1987] M.A. Fischler and H.C. Wolf. Linear Delineation. In *Readings in Computer Vision (M.A. Fischler and O. firschein, eds.)*, pages 204–209. Morgan Kaufmann, 1987.

[Fischler *et al.*, 1981] M.A. Fischler, J.M. Tenenbaum, and Wolf H.C. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. *CGIP*, 15(3):201–223, March 1981.

[Fischler, 1997] M.A. Fischler. Finding the perceptually obvious path. In *DARPA97*, pages 957–970, 1997.

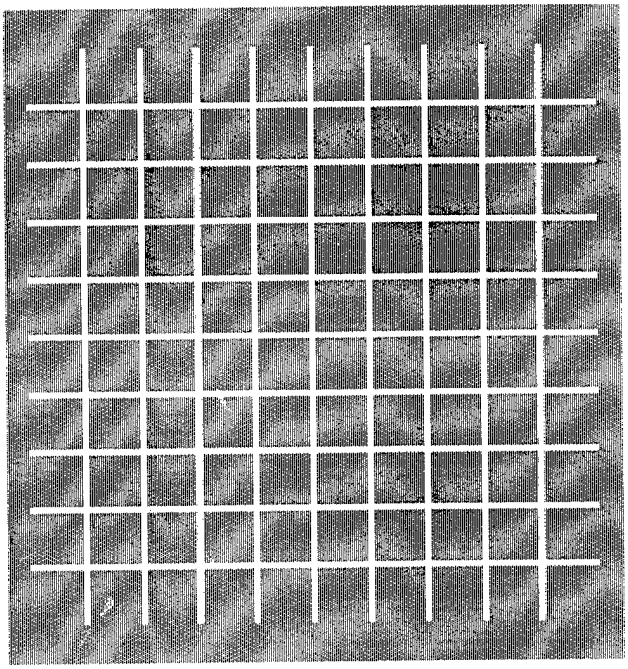


Figure 2: Grid-Test-Pattern

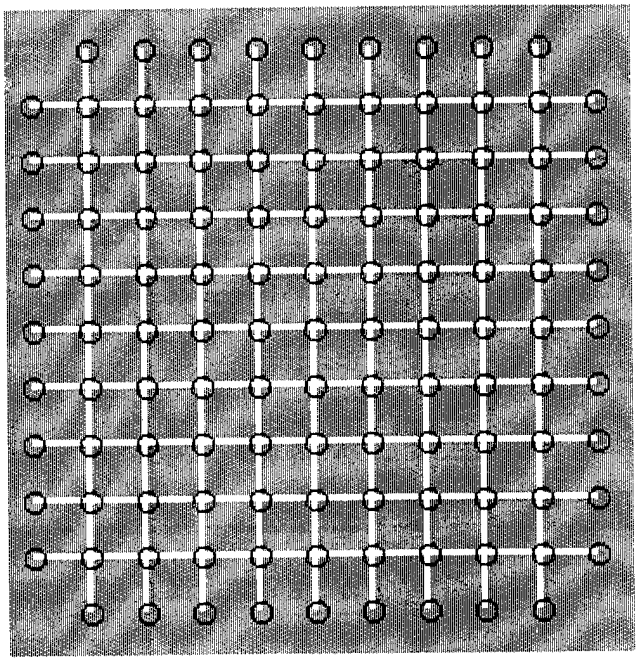


Figure 3: SNL result (*road-segs2*) for Grid-Test-Pattern

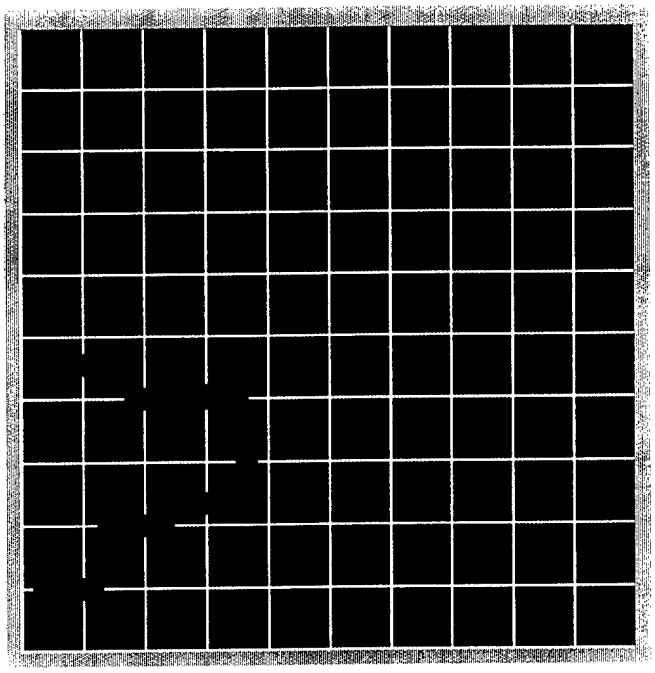


Figure 4: Grid-Test-Pattern-Random-Erase

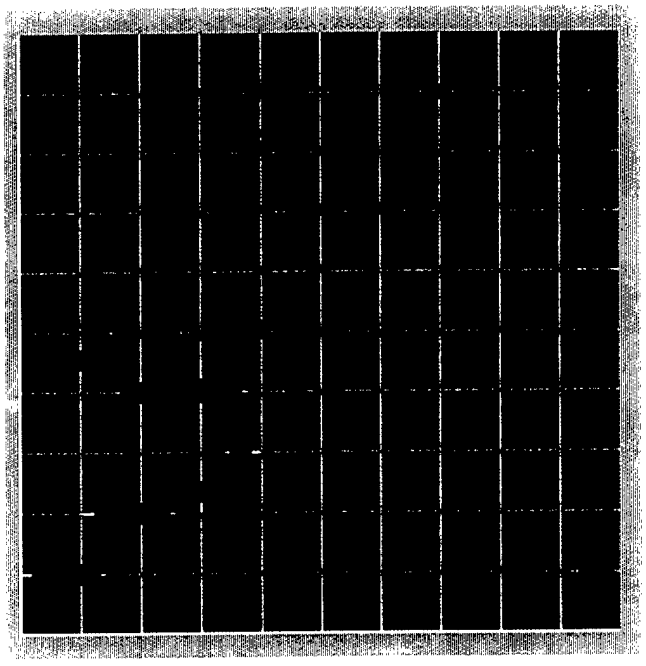


Figure 5: SNL input (*road-segs*) for grid-test-pattern-random-erase.

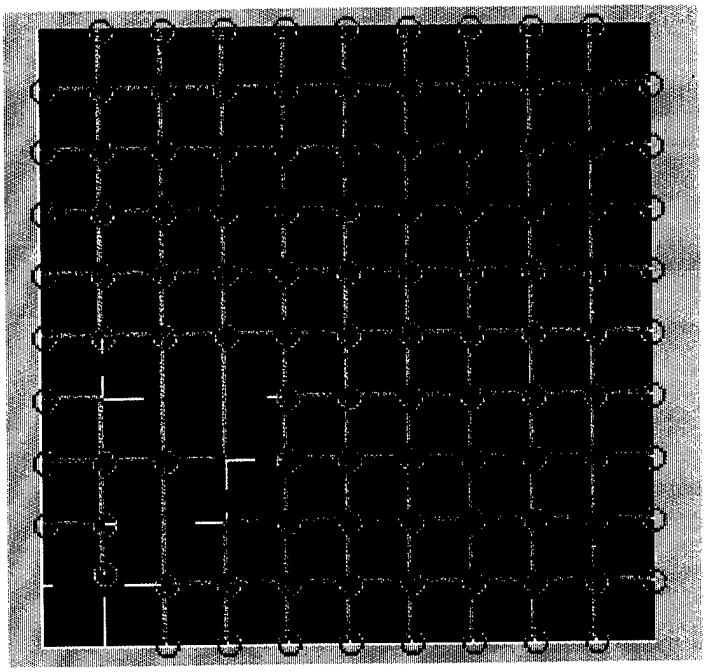


Figure 6: SNL result (*road-segs2*) for grid-test-pattern-random-erase.

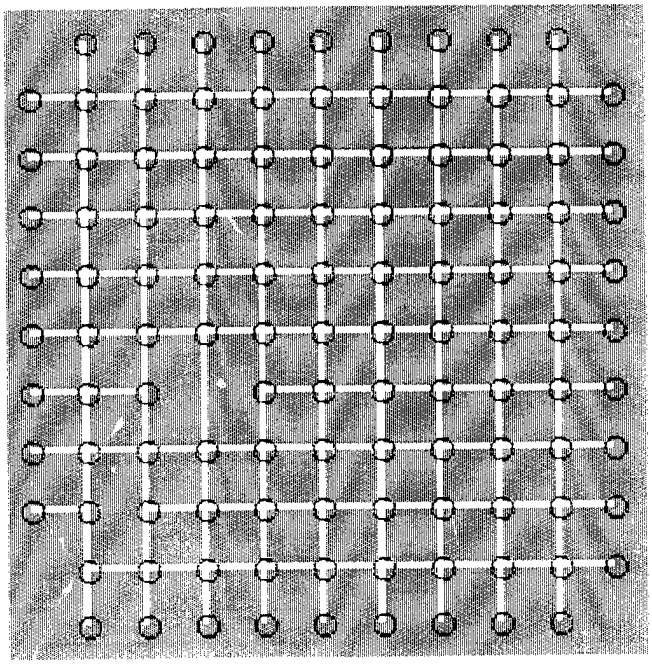


Figure 7: augmented-SNL result (*road-segs2*) for grid-test-pattern-random-erase.

Figure 8: Ft. Irwin

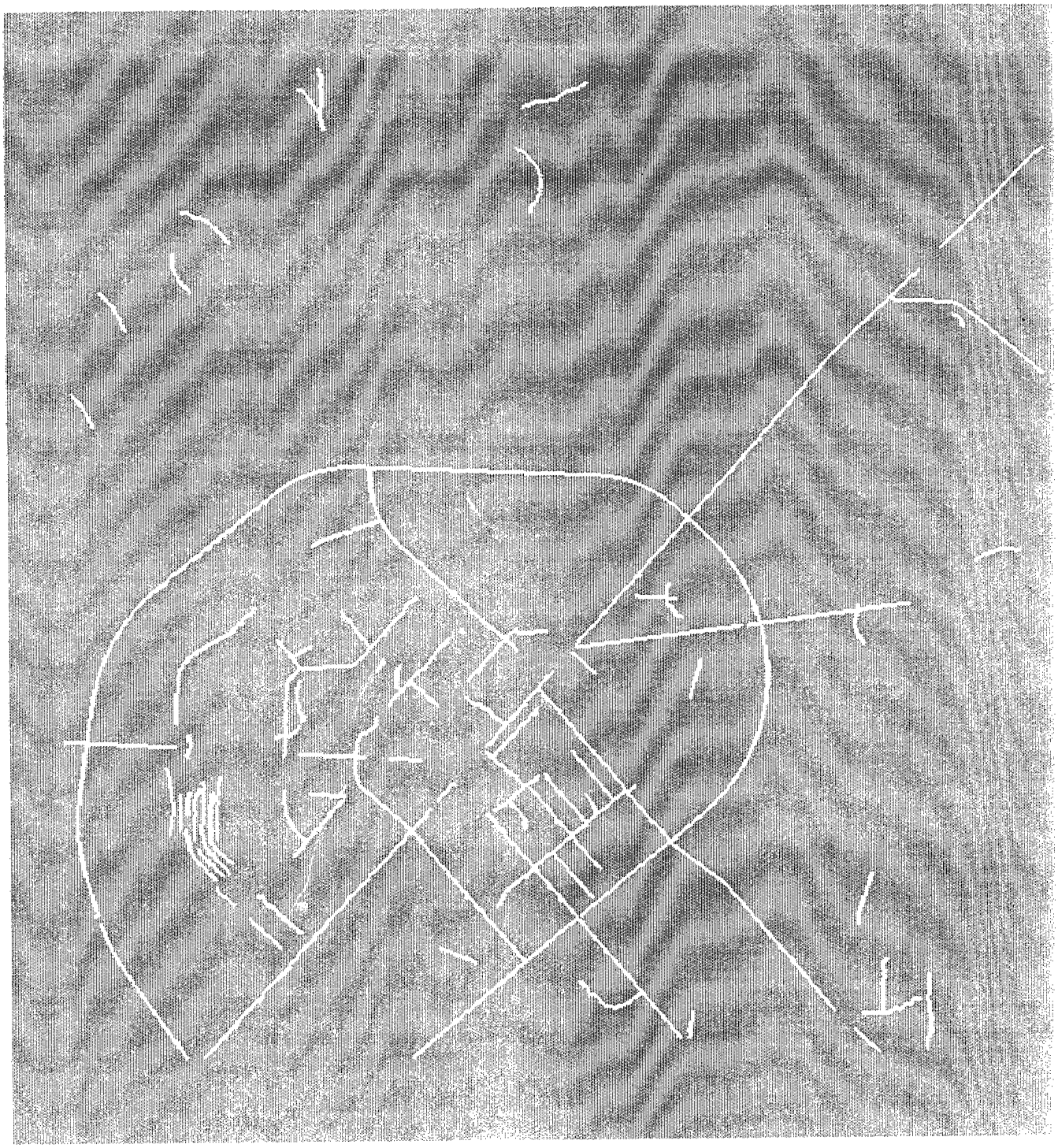


Figure 9: Ft Irwin *road-segs*

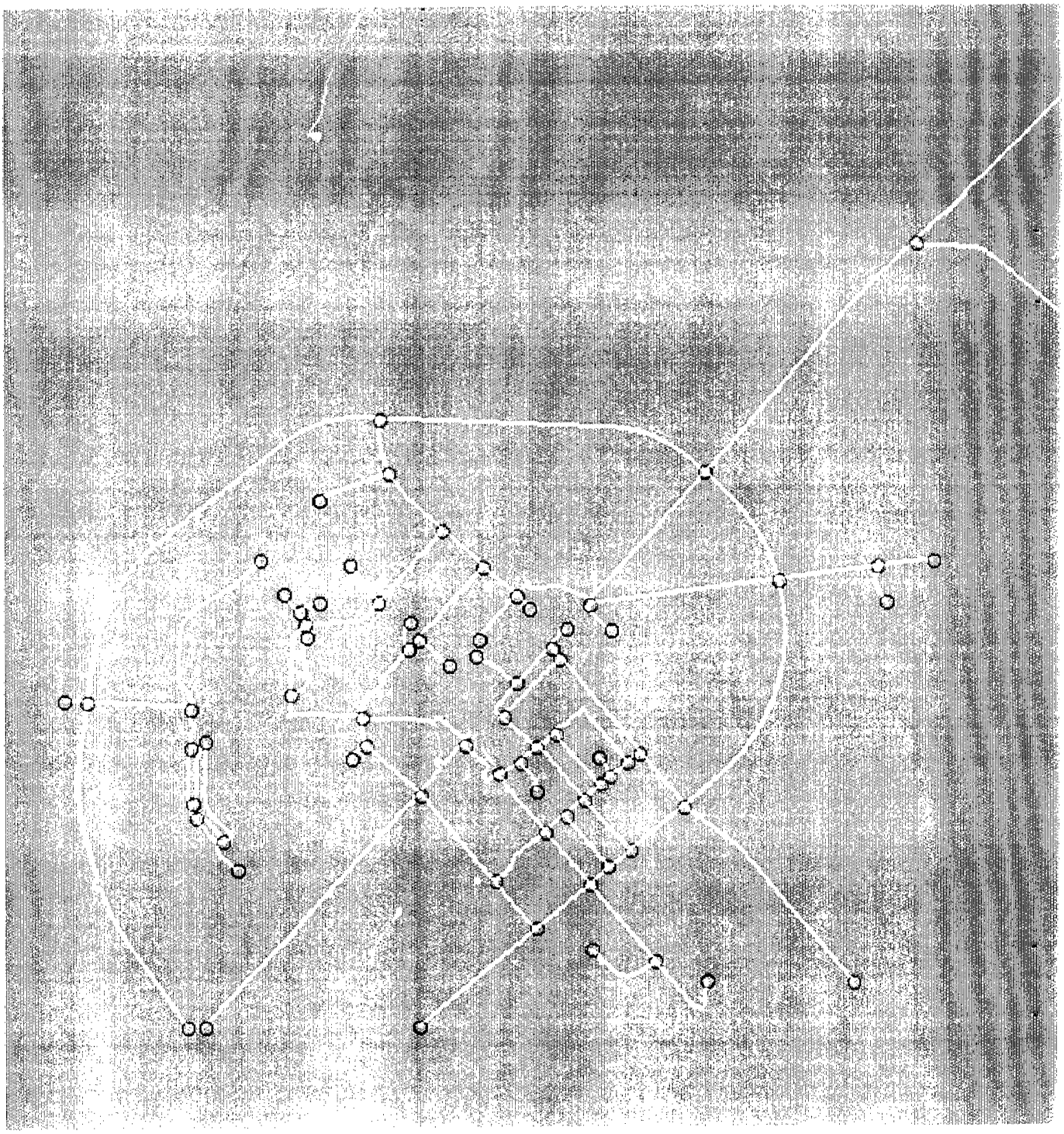


Figure 10: Ft Irwin (*road-segs2*) Road-Model compiled by basic-SNL

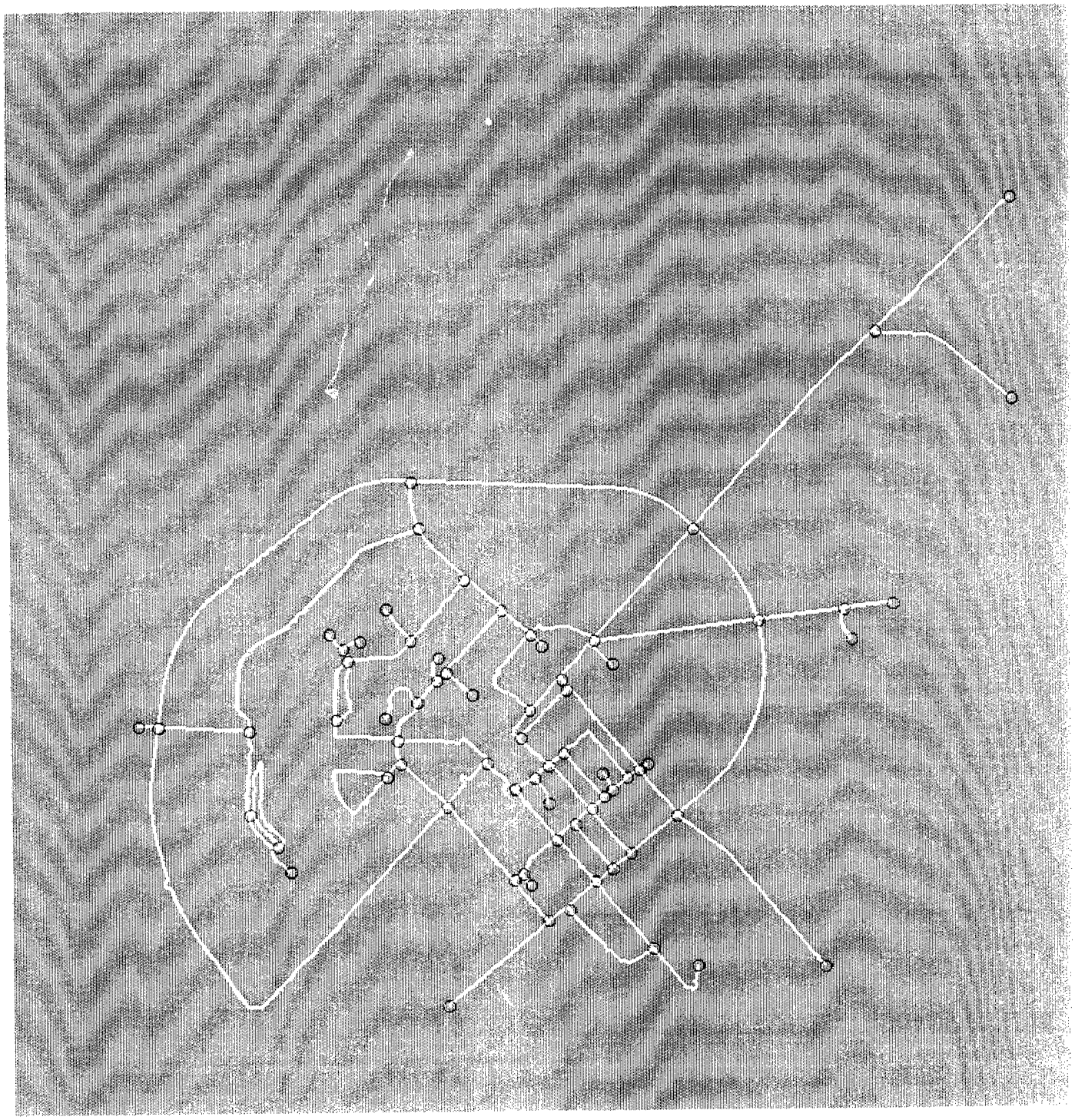


Figure 11: Ft Irwin Road-Model compiled by extended-SNL

Figure 12: ALV-Image-Window

Figure 13: ALV-Image-Window: *rpaths* overlay

Figure 14: ALV-Image-Window; *road-segs* overlay

Figure 15: ALV-Image-Window; *road-segs2* using basic-SNL

Figure 16: ALV-Image-Window; *road-segs*(yellow) and added basic-SNL-links(red)

Figure 17: ALV-Image-Window; basic-SNL-linked-network-components(colored) over *road-segs*(yellow)

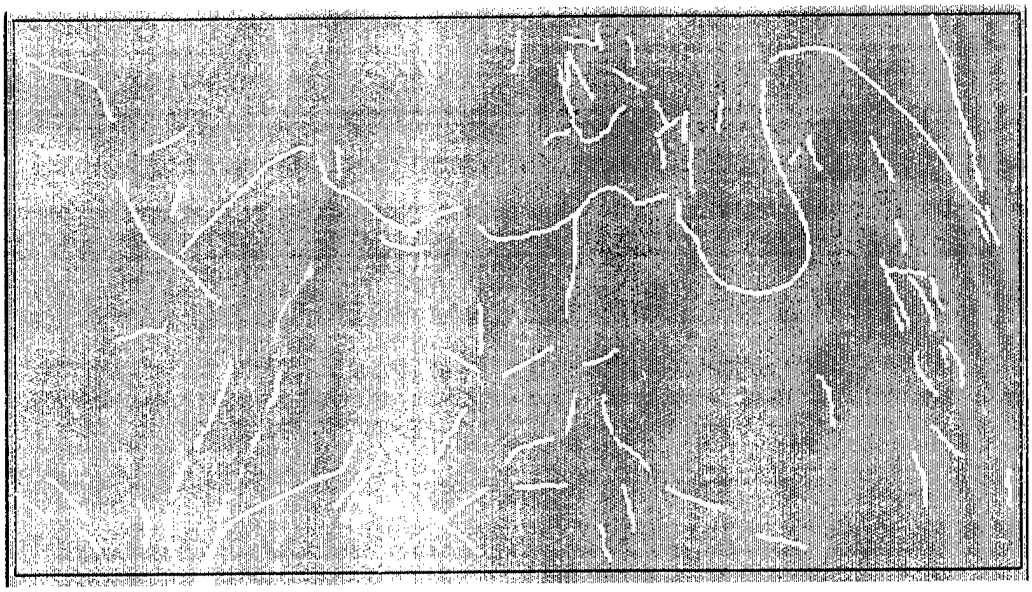


Figure 18: ALV-Image-Window; *road-segs*

Figure 19: ALV-Image-Window

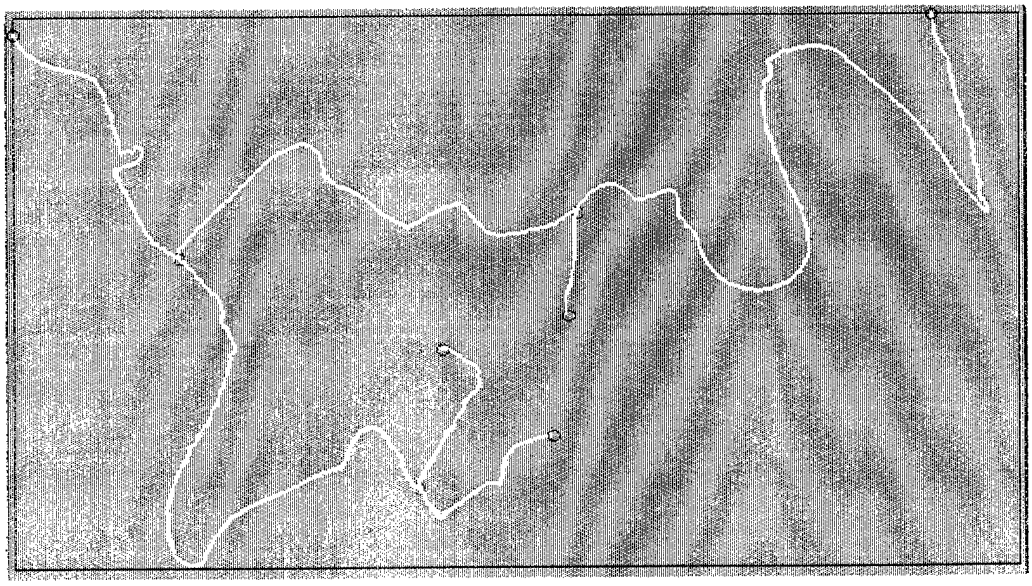


Figure 20: ALV-Image-Window; *road-segs2*

APPENDIX 3:

Visual Similarity, Judgmental Certainty, and Stereo Correspondence

Visual Similarity, Judgmental Certainty and Stereo Correspondence *

James Ze Wang† Martin A. Fischler‡
Artificial Intelligence Center, SRI International, Menlo Park, CA 94025

Abstract

Normal human vision is nearly infallible in modeling the visually sensed physical environment in which it evolved. In contrast, most currently available computer vision systems fall far short of human performance in this task, and further, they are generally not capable of being able to assert the correctness of their judgments. In computerized stereo matching systems, correctness of the similarity/identity-matching is almost never *guaranteed*. In this paper, we explore the question of the extent to which judgments of similarity/identity can be made essentially error-free in support of obtaining a relatively dense depth model of a natural outdoor scene. We argue for the necessity of simultaneously producing a crude scene-specific semantic "overlay". For our experiments, we designed a wavelet-based stereo matching algorithm and use "classification-trees" to create a primitive semantic overlay of the scene. A series of mutually independent filters has been designed and implemented based on the study of different error sources. Photometric appearance, camera imaging geometry and scene constraints are utilized in these filters. When tested on different sets of stereo images, our system has demonstrated above 98% correctness on *asserted* matches. Finally, we provide a principled basis for relatively dense depth recovery.

1 Introduction

Vision, by animals or machines, is an inductive process which results in the construction of models, or theories, about the sensed environment. Unlike mathematical assertions, with respect to which one can make absolute judgments about correctness (actually, only about consistency with some assumed set of axioms), any assertion about the physical world can only be disconfirmed – never established with certainty. Never the less, our introspection and experience assures us that normal human vision is almost infallible in modeling the visually sensed physical environment in which we evolved and with which we directly interact. It is almost never the case that there is a *hole* in our visual field where our visual system can't produce an instantiated model, and it is very rare that our visually produced models cause us to fail in some task because they were *incorrect*. Even in the case of illusions, it is not obvious that our visually guided behavior would suffer from the same errors our conscious introspection is subject to. (Obviously, geometric modeling becomes less reliable as the distance from the sensor increases.)

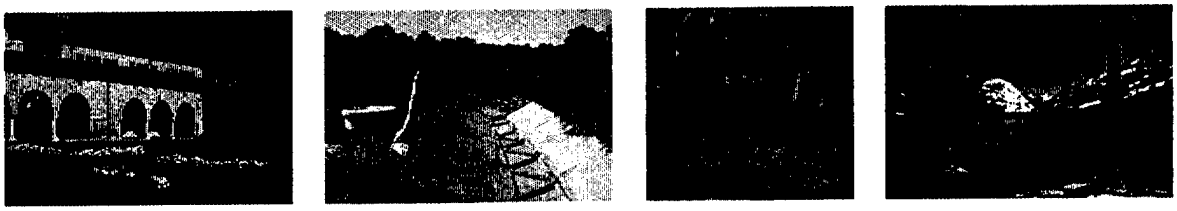
In contrast, most currently available computer vision systems fall far short of human performance in this task, and additionally, they make no attempt, or are generally not capable of being able to assert the correctness of their judgments in proposing correspondences required for dense stereo depth modeling. In computerized stereo matching systems correctness of the similarity/identity matching is almost never *guaranteed*. There are some important exceptions, especially in regard to "structure-from-motion" problems

*This work was sponsored by the Defense Advanced Research Projects Agency under contract DACA76-92-C-0008 monitored by the U.S. Army Topographic Engineering Center, Alexandria, VA. 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 Advanced Research Projects Agency, the United States Government, or SRI International. Original figures in this paper are located at the URL: `http://www-db.stanford.edu/~wangz/project/stereo/J00/`

†Also of Department of Computer Science and Department of Medical Informatics, Stanford University, Stanford, CA 94305. Email: `wangz@cs.stanford.edu`

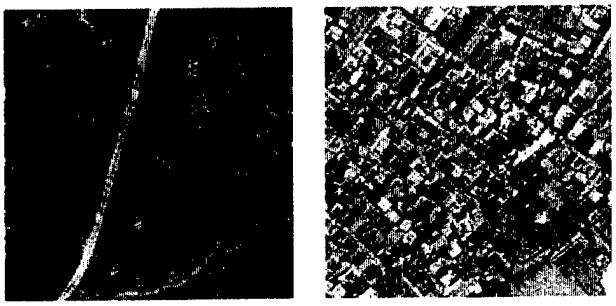
‡Email: `fischler@ai.sri.com`

where efforts are made to either statistically predict and verify the accuracy of the 3-D registration methods [24, 7, 20, 25, 13, 17, 21, 26] or to select correspondences from a predetermined set that are consistent with a "rigid" spatial configuration.



Stanford Main Quad | Stanford Hill | Stanford Trees | Yosemite Lake

Figure 1: **Natural outdoor scenes used for our experimental investigation.**



Ft. Benning | Kawasaki

Figure 2: **Aerial imagery used for our experimental investigation.**

In this paper, we explore the question of the extent to which judgments of similarity/identity (believed to be the bias of human stereopsis) can be made *essentially* error-free in the context of stereo matching in the natural outdoor world. And further, how such a (possibly sparse) set of correspondences could provide a dense depth model.

The paper is organized as follows. In Section 2, we discuss the key problems to be addressed and our approach to their solution. Section 3 describes our experimental environment. The details of our matching algorithm are given in Section 4. Section 5 presents experimental results on real-world image data. Section 6 discusses the experimental results. Finally, Section 7 presents conclusions and future directions.

2 The Central Problems

Human intelligence would be relatively worthless in a non-causal world. To exploit causality, it is necessary to be able categorize and recognize objects and events, in order to predict what will happen next or to take appropriate action based on past experience.

In machine vision, the categorization problem is central and pervasive. In this paper we examine one of the simplest instances of this problem – the problem of establishing stereo correspondence – and address the key question: *How can one be "certain" that a stereo match is correct.*

In order to answer this question, and exploit the answer, we address the following issues: what is visual similarity/uniqueness and how can we measure it; what is judgmental certainty and how can it be established; what is the role of semantic scene understanding in judgments about stereo correspondence.

2.1 Visual Similarity

A similarity metric for assigning distinct objects membership in a classification scheme can be completely arbitrary and is almost certain to be problem dependent. For example, we would not expect the metric used

for classifying/recognizing flaws in a printed circuit board to be the preferred metric for correctly classifying images of trees according to species. Even when we restrict similarity judgments to the *identity* classes of real 3-D world objects (the distinct objects themselves, as opposed to class membership(s) of these distinct objects), there is a large set of alternative metrics that depend on how we define (or can acquire) our available observations, what we mean by an *object*, and how we intend to use the answer. For example, if we recognize the front and rear views of the same person in two different images, this could be useful for some purposes but relatively worthless for geometric recovery via stereo correspondence. Thus, any meaningful discussion of matching and the corresponding quantification of "degree of similarity" must be *grounded* in a specific problem. We use stereo vision as the grounded reference for evaluating our contribution. In this regard, we wish to understand and duplicate human stereo *competence*, but not necessarily the explicit mechanisms employed by the HVS.

We note that the human visual system operates in real-time, *below* the conscious level, to produce a 3-D representation of the environment. It is reasonable to assume that stereopsis is pre-attentive. This would normally imply that it uses little or no scene-specific contextual knowledge in arriving at its instantaneous judgments, but follows a preselected procedure (or algorithm). We will argue that effective stereo in the outdoor world must involve scene-specific context. Thus, a *solution* to the problem of designing an *infallible* stereo machine cannot be based solely on comparing the intensity variations in two (or more) images.

2.2 Judgmental Certainty

The problem that the HVS appears to have solved, the ability to make uniformly correct judgments in an uncertain world, is a core problem we address in this paper. There are, essentially, only two ways of judging when a fallible process has produced a correct answer:

1. Apply some known criterion/condition or test for correctness (that may not be competent in itself to obtain the desired answer). In mathematics we might not know how to prove a given theorem, but we know how to check a proof when offered one (regardless of the reliability of the source of the proof).

2. Get the *opinions* of a suitably sized collection of *informed independent* sources, and accept the proposed solution only when there is both a sufficient *consensus* of agreement, *and* when additional criterion for a valid model are satisfied: the additional criterion include stability (e.g., the derived model does not change in a significant way under *small* perturbations of the data or the viewing conditions) and limited model *complexity* (given too many free variables in a model, it can be made consistent with any collection of data).

In this paper we focus on method (2) for establishing judgmental certainty. The application of this approach to problems in vision requires a careful examination of what is meant by the terms *informed* and *independent* in the vision context.

In its most fundamental sense, by **independent** opinions we mean that the errors made by the sources of these opinions, with regard to some given problem, are uncorrelated.

By **informed**, we mean (at least) that a process is more likely than pure chance to produce a correct answer. We will show later that an opinion can only be informed relative to some specific collection of error types/conditions. In particular, we must ultimately be concerned with:

- incorrect assumptions
- an incomplete model (e.g., some key variables are omitted – such as lens distortion in the context of a perspective imaging model)
- incomplete set of observations/information
- incorrect observations/information
- approximations (e.g., due to the finite resolution of measuring devices, and also, the representation of continuous numerical quantities in a machine)
- incorrect implementation (e.g., nerve damage, programming errors)

- probabilistic algorithms or a guessing strategy (errors are expected)
- an inappropriate utility function

Some of the corresponding visual phenomena include: (1) occlusion (2) ambiguity (3) distortion (4) incorrect assumptions about (or approximations with respect to) reflectance, surface continuity, camera geometry, illumination (5) computational errors or numerical instability in computing optical or geometric transforms.

2.3 Three Primary Information Sources for Image-Based Scene Modeling

We consider three primary information sources for image-based geometric scene modeling: (1) the image(s): photometric appearance and shape (2) the camera(s): imaging geometry constraints (3) the scene: scene domain and scene-specific constraints such as physical, semantic, geometric, photometric relationships and regularities.

2.3.1 Photometric Appearance-Based Similarity

From a statistical/signal-processing point of view, the objects of interest can be characterized using an attribute-vector of measurements made on the objects, and we then quantify the similarity relationship between two objects by the "normalized" distance between their attribute vectors. We note that even correlation-based matching can be viewed in this way – here the attribute vector is the ordered set of intensity values in the *correlation* patch. Never the less, it is difficult to deal with certain types of similarity problems using this formalism. In particular, line drawings cannot be well described this way, and more important, the local image appearance of (say) grass or other types of *nearby* vegetation is highly unstable to small shifts in viewing position. While we question the adequacy of vector-space characterization as the sole basis for natural outdoor scene stereo matching/modeling there are very few other practical alternatives available at present.

2.3.2 Imaging-Geometry Based Constraints on Feature Matching

Advances made over the past two decades in projective geometry and robust statistical estimation [15, 5, 18, 1, 16, 12, 10], appear to provide a relatively complete basis for exploiting imaging geometry in both depth recovery and in rejecting point correspondences that are inconsistent with the derived camera model. In this paper, we have little to add in this area. However, we do employ projective constraints beyond those directly associated with camera modeling. For example, We have implemented a filter that uses a plane-to-plane linear transform to reject errors associated with semantically identified planar scene features.

And, of course, we do not wish to imply that additional advances are not needed in this area. We note that the HVS is not completely dependent on a projective model of the imaging process – it can recover a "qualitative" geometric model of a scene from highly distorted images.

2.3.3 Scene Based Constraints on Feature Matching

It is almost universally the case that stereo-based depth-recovery systems are designed to operate without reference to scene semantics. In the case of the HVS, it is commonly assumed that stereopsis occurs very early in the visual processing chain, is pre-attentive, and is based purely on some form of *local matching*. Julesz [8] has shown that stereopsis can occur in the absence of any meaningful information in the individual images of a stereo pair. Never the less, we argue in this paper that stereo depth recovery in the natural outdoor world must invoke scene-specific semantic knowledge to create a relatively dense meaningful depth-model. For example, in the Tenaya Lake picture (Figure 10) most of the scene is composed of either sky or lake. The lake is especially interesting in that it can appear as a large *mirror-like* surface. Reflected objects can be matched to produce a depth map which is consistent over a large number of views, but which is incorrect. Under circumstances where the the water surface is refractive rather than reflective, we can again form valid matches which produce incorrect depth measurements (if we make the usual assumption that light travels in straight lines). On the other hand, if we know we are looking at a large flat body of

water, we could profit from its known planar geometry to constrain matching of objects on its immediate boundary, and to obtain a correct depth model (via interpolation) for its surface. We can even correct for its refractive properties if we need to estimate its depth. In the case of the sky, we might determine that it is homogeneous, and thus not suitable for matching, without knowing what it is. However, where the sky is visible through the tree foliage, a purely geometric system might well try to interpolate depth from the surrounding valid matches – this works for the lake (which might also appear photometrically homogeneous) but is obviously incorrect for sky patches. There are a large number of similar considerations (e.g., fire, smoke, snow, insubstantial surfaces – such as grass or foliage, ...) that force one to conclude that some form of crude semantic overlay must be available to support a depth recovery system whose results must be reasonably complete and correct. We have previously developed techniques that could be used to compute a suitable semantic overlay [6], but for most of the experiments described in this paper, we employed a method based on recent work using classification-trees [2].

3 The Experimental Environment

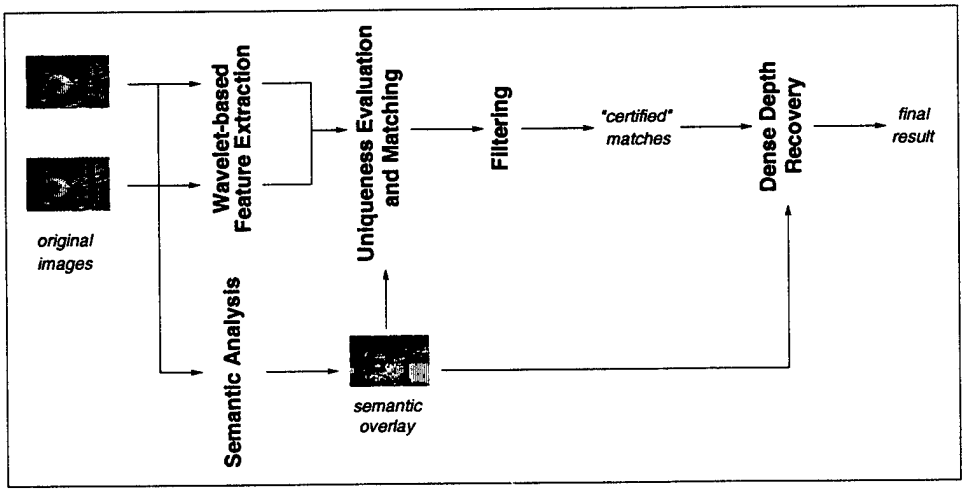


Figure 3: **Basic structure of the current experimental system.**

We assume that the images to be processed were obtained with a stereo camera configuration similar to the HVS. Two essentially identical cameras (or a single camera) that is used to view the scene at approximately the same time from two closely spaced locations. The cameras have vertically oriented image-planes (approximately) parallel to each other. The two images of a stereo pair should be quite similar to each other modulo some projective and lens distortion, a horizontal shift in scene content, some differences in occluded regions, and some intensity variation due to film processing and non-Lambertian reflective surfaces in the scene. The currently implemented experimental configuration was intended for ground-level images of natural outdoor scenes; it was not intended to model scenes with man-made objects or aerial views. However, we did apply it to aerial imagery and outdoor scenes with man-made objects. Figures 1 and 2 show some of the images we have used in our experiments.

Our goal in this experimental study was not the implementation and testing of the complete stereo system we envision, but rather to demonstrate that we can can extract a set of correct matches with a specified maximum percentage of errors in each uncontrived stereo pair we process and then show that based on such a sampling of "known correct" matches, and a "semantic overlay" constructed (nominally) in parallel with the matching results, we can obtain a dense depth map that is superior to conventional (2-image) stereo models. Some of the components and processing steps in our experimental work were chosen for convenience and accessibility, rather than reflecting the ideal design.

In order to compare our results to existing state-of-the art stereo/matching systems, and to illustrate the importance of the concepts proposed in our paper, we took advantage of an excellent publicly accessible

`image-matching` algorithm[1] which implements a robust technique for binocular image matching by exploiting the epi-polar constraint. It uses correlation and relaxation methods to find an initial set of matches, and then use the Least Median of Squares technique to discard false matches. (We realize that INRIA's latest developments (e.g. [17]) on stereo matching are not necessarily included in the available software.)

A pervasive problem in stereo/matching research is the evaluation of results obtained from experiments using real images, and especially when the data is ground-level imagery of natural outdoor scenes. Some of our earlier evaluations in this effort were based on a "manual" assessment of each asserted match-pair. Our more recent experiments (most involving aerial imagery) exploited some new evaluation ideas and techniques [11] which do not require the availability of "ground truth." The basic idea is that if we can acquire three or more calibrated images that cover the same area, and we can find a common point appearing in two or more asserted matches (across three or more images), then a necessary condition that this set of matches are all correct is that they all "recover" the same ground point. While in a real application we could use the evaluation software to eliminate errors when three or more views were available, for the purposes of this paper, we always restricted our input data to the algorithm to be a single stereo image-pair.

4 The Matching Algorithm

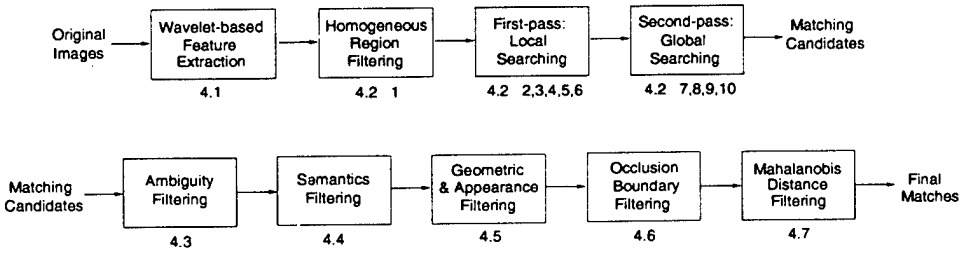


Figure 4: **The main steps in the matching algorithm.**

The current experimental stereo configuration consists of several modules: a wavelet-based feature extraction module, a semantic analysis module, a uniqueness evaluation and matching module, a filtering module and an interpolation module for dense depth recovery. Figure 3 shows the basic structure of the system. Figure 4 shows the computational flow of our matching algorithm.

In this section, we provide the details of the matching algorithm. The algorithm takes two images as input. It can also take advantage of a precomputed fundamental matrix if available.

Assume an image is specified by a set of pixels $\mathcal{I} = \{(i,j),\ i = 0,...,m-1,\ j = 0,...,n-1\}$. We denote the pair of images to be matched as $\mathcal{I}_1$ and $\mathcal{I}_2$. We crop and process the images so that they are of the same size, $m \times n$, and of roughly the same brightness. If the 3×3 fundamental matrix for the pair of images is given, we denote it to be F.

4.1 Wavelet-based Feature Extraction

Various experiments [27, 22, 23] have shown that Daubechies' wavelets [3, 4, 14, 9] are well suited for characterizing localized information in natural signals such as sounds and images. We characterize the local intensity information at each pixel location in each image with a vector of seven wavelet coefficients, i.e. one low frequency coefficient and six high frequency coefficients, obtained from framed wavelet transforms.

1. Apply the Daubechies-4 wavelet filter to each row of the image. We obtain a low-pass vector of length n and a high-pass vector of length n for each row. For each original image, we obtain two matrices of coefficients, each having the same dimensions as the original image. We denote these matrices $L_{\mathcal{I}_1}$ and $H_{\mathcal{I}_1}$ for the first image, and $L_{\mathcal{I}_2}$ and $H_{\mathcal{I}_2}$ for the second image.

[1] Available from INRIA at: `http://www.inria.fr/robotvis/personnel/zzhang`

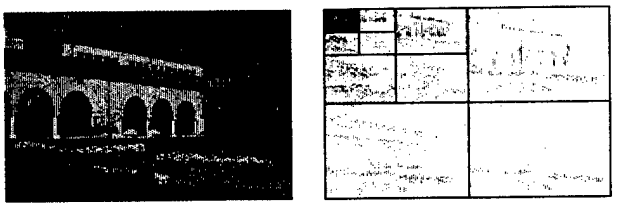


Figure 5: **A normal 3-level wavelet transform.**

2. *Without* down-sampling, transpose the four matrices, $L_{\mathfrak{I}_1}$, $H_{\mathfrak{I}_1}$, $L_{\mathfrak{I}_2}$ and $H_{\mathfrak{I}_2}$. This step is different from the traditional wavelet transform where a down-sampling is performed.

3. Apply the Daubechies-4 wavelet filter to each row of the four transposed matrices. We obtain a low-pass vector of length m and a high-pass vector of length m for each row. For each of these matrices, we obtain two matrices of coefficients, each having the size $n \times m$. We denote these matrices $LL_{\mathfrak{I}_1}$, $LH_{\mathfrak{I}_1}$, $HL_{\mathfrak{I}_1}$, and $HH_{\mathfrak{I}_1}$, for the first image, and $LL_{\mathfrak{I}_2}$, $LH_{\mathfrak{I}_2}$, $HL_{\mathfrak{I}_2}$, and $HH_{\mathfrak{I}_2}$, for the second image.

4. *Without* down-sampling, transpose the eight matrices. Now we get eight matrices of the same size, $m \times n$. Again, this step is different from the traditional wavelet transform. Figure 5 shows a normal 3-level wavelet transform. Figure 6 shows the notations.

5. Apply Step 1 to Step 4 on the matrices $LL_{\mathfrak{I}_1}$ and $LL_{\mathfrak{I}_2}$ to obtain an additional four matrices for each one of them. Now we have decomposed each original image into seven matrices of distinct frequency bands, three from previous steps and four from this step.

6. Shift the matrices in both dimensions for 4 pixels so that the coefficients in the matrices correspond to the actual location of the pixels in the original image. During the image matching process, we avoid matching in the border area due to the boundary problem with the wavelet filtering.

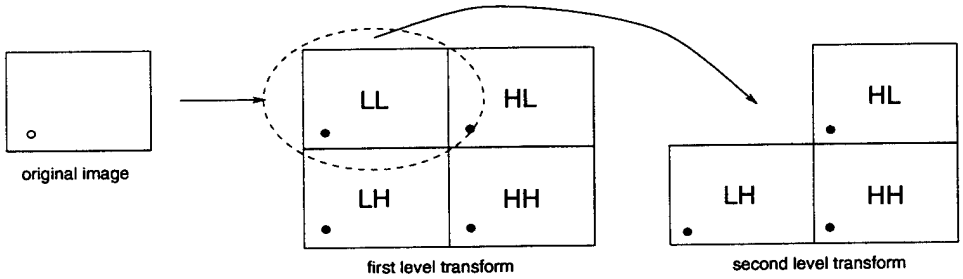


Figure 6: **Forming the feature vector.**

7. For each pixel in the original images, we collect the corresponding coefficients in the seven matrices to form a feature vector of seven dimensions. We denote these 7-dimensional vectors as $\mathfrak{V}_{\mathfrak{I}_1(i,j)}$ and $\mathfrak{V}_{\mathfrak{I}_2(i,j)}$. Figure 6 illustrates the process.

4.2 Uniqueness Evaluation

Correct stereo matching requires an evaluation of both uniqueness and similarity. We first evaluate the uniqueness of the wavelet descriptor at each pixel location in the image. Denote A as the matrix containing the *ambiguity scores* of the *pixels* in the image. The size of A is $m \times n$.

In this step, we do not restrict the search to a single epi-polar line. (Stevenson [19] has shown that human stereo matching also is not restricted to epi-polar lines.)

If the fundamental matrix F is given, we perform the following computation.

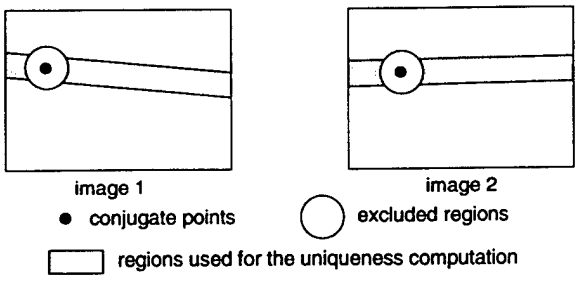


Figure 7: **Uniqueness evaluation.**

1. We exclude homogeneous regions by checking the high frequency wavelet coefficients in the feature vectors. If the coefficients for a particular point are too small, we do not consider that point as a match candidate.

2. Initialize the matrix A to a zero matrix.

3. Set two constants s_1 and s_2 to *small* values (e.g. 3-5).

4. For each pixel $\mathcal{I}_1(i_1, j_1)$ in the left image, use the fundamental matrix F to find the corresponding band of width $2s_1$ adjacent epi-polar lines of pixels in the right image.

5. For each pixel within the band of $2s_1$ epi-polar lines, denoted as $\mathcal{I}_2(i_2, j_2)$, compute the Euclidean distance between the feature vectors $\mathcal{V}_{\mathcal{I}_1(i_1,j_1)}$ and $\mathcal{V}_{\mathcal{I}_2(i_2,j_2)}$. Denote the distance as $d(i_1, j_1; i_2, j_2)$.

6. Sort the distances $d(i_1, j_1; i_2', j_2')$, where (i_2', j_2') run over all the pixels in the band of $2s_1$ adjacent epi-polar lines. Find the closest match to $\mathcal{I}_1(i_1, j_1)$ and denote it as $\hat{\mathcal{I}}_2(i_2, j_2)$. That is, $d(i_1, j_1; i_2, j_2)$ is the minimum over all d's. We call the pair of points as a conjugate pair.

7. For all pixels $\mathcal{I}_2(i_2', j_2')$ in the second image such that $(i_2' - i_2)^2 + (j_2' - j_2)^2 < {s_2}^2$, we compute the maximum $d(i_2, j_2; i_2', j_2')$. Denote the maximum as t_2.

8. For all pixels $\mathcal{I}_2(i_2', j_2')$ in the pixel band in the second image such that $(i_2' - i_2)^2 + (j_2' - j_2)^2 > {s_2}^2$, if $d(i_1, j_1; i_2', j_2') < t_2$ holds, we discard the match. Otherwise,

$$A(i_1, j_1) = A(i_1, j_1) + \frac{1}{d(i_1, j_1; i_2', j_2')} \quad .$$

9. For all pixels $\mathcal{I}_1(i_1', j_1')$ in the first image so that $(i_1' - i_1)^2 + (j_1' - j_1)^2 < {s_2}^2$, we compute the maximum $d(i_1, j_1; i_1', j_1')$. Denote the maximum as t_1.

10. For all pixels $\mathcal{I}_2(i_1', j_1')$ in the pixel band in the first image such that $(i_1' - i_1)^2 + (j_1' - j_1)^2 > {s_2}^2$, if $d(i_1', j_1'; i_2, j_2) < t_1$ holds, we discard the match. Otherwise,

$$A(i_1, j_1) = A(i_1, j_1) + \frac{1}{d(i_1', j_1'; i_2, j_2)} \quad .$$

We note that t_1 and t_2 are computed thresholds on acceptable uniqueness. Figure 7 shows that the regions around the conjugate pair are excluded for the uniqueness evaluation. If the fundamental matrix F is not given as an input, we use a band of $2s_1$ rows of pixel around the point in the first image to determine the uniqueness of the point.

We now have an ambiguity score matrix A for the image pair. During the matching phase, we require that the components of a match pair is not only similar but also unique.

4.3 Initial Matching

In this part of the process, we try to find a list of about N conjugate pairs that satisfy criteria for both uniqueness and similarity. For our applications, we set N to a (nominal) value of 300.

If the fundamental matrix F is given, we perform the following procedure.

1. Without considering the homogeneous regions, sort values in the ambiguity matrix A.
2. Set $1 \rightarrow c$.
3. If $c > N$, terminate.
4. For each pixel in the left image with the next smallest ambiguity score in A, denoted as $\hat{\mathcal{I}}_1(i_1, j_1)$, compute the global ambiguity score over the entire right image.
5. If the global ambiguity score is smaller than a threshold determined similar to t_2, we use the Euclidean distance between the feature vectors $\mathcal{V}_{\mathcal{I}_1(i_1,j_1)}$ and $\mathcal{V}_{\mathcal{I}_2(i_2',j_2')}$ to find a best match within the adjacent $2s_1$ epi-polar lines in the right image. Denote the match as $\hat{\mathcal{I}}_2(i_2, j_2)$.
6. Use the Euclidean distance between $\mathcal{V}_{\mathcal{I}_1(i_1',j_1')}$ and $\mathcal{V}_{\mathcal{I}_2(i_2,j_2)}$ to find a best match within the adjacent $2s_1$ epi-polar lines in the left image. Denote it $\hat{\mathcal{I}}_1(i_3, j_3)$.
7. If $i_3 - i_1 > 1$ or $j_3 - j_1 > 1$, we discard the match.
8. Find the best match of $\mathcal{I}_1(i_1, j_1)$, denoted $\hat{\mathcal{I}}_k(i_4, j_4)$, within the entire two images except the neighborhoods of the points $\mathcal{I}_1(i_1, j_1)$ and $\mathcal{I}_2(i_2, j_2)$. Denote the Euclidean distance between $\mathcal{I}_1(i_1, j_1)$ and $\mathcal{I}_k(i_4, j_4)$ as d_1.
9. Find the best match of $\mathcal{I}_2(i_2, j_2)$, denoted $\hat{\mathcal{I}}_l(i_5, j_5)$, within the entire two images except the neighborhoods of the points $\mathcal{I}_1(i_1, j_1)$ and $\mathcal{I}_2(i_2, j_2)$. Denote the Euclidean distance between $\mathcal{I}_1(i_1, j_1)$ and $\mathcal{I}_l(i_5, j_5)$ as d_2.
10. If $d(i_1, j_1; i_2, j_2) < d_1$ and $d(i_1, j_1; i_2, j_2) < d_2$, accept the match $\mathcal{I}_1(i_1, j_1)$ and $\mathcal{I}_2(i_2, j_2)$ as a valid conjugate pair for the initial matching stage. Otherwise, discard the match.
11. Set $c + 1 \rightarrow c$. Go to Step 3.

We have now obtained on the order of 300 conjugate pairs, where each pair satisfies the following condition: each member of a pair has only one other potential match in the set of unique points, and this single match is its conjugate in the other image.

If the fundamental matrix F is not given as an input, we use a band of $2s_1$ rows of pixel around the point in the images for the computation.

4.4 Semantic Overlay Filtering

If the scene is a ground-level natural out-door scene, we create a rough semantics overlay to eliminate matches in regions (e.g. sky and water) where we have little confidence of finding correct conjugate pairs. If the scene is not a natural out-door scene, we skip this filter step.

We derive a rough semantic overlay of each image of the stereo pair. For most of the experiments discussed in this paper, we use training samples from a few scenes similar (but distinct) from the experimental scenes to create a decision tree structure using the classification and regression trees (CART) algorithms [2]. CART, developed by Breiman et al., has been widely used in computer-aided clinical diagnosis research. Figure 8 shows the structure of a classification tree generated by the CART algorithm.

In our experiments, we used a sequence of seven training images representing *sky, stone, river/lake, grass* and *tree/forest.* Figure 9 shows five of the seven training images. We use the mean colors and variances of 4×4 blocks in RGB color space as the components of the training feature vector. These features are simple but appear capable of distinguishing the above five classes. For gray scale images, we use only the mean intensity and variance of 4×4 blocks as the components of the training feature vector.

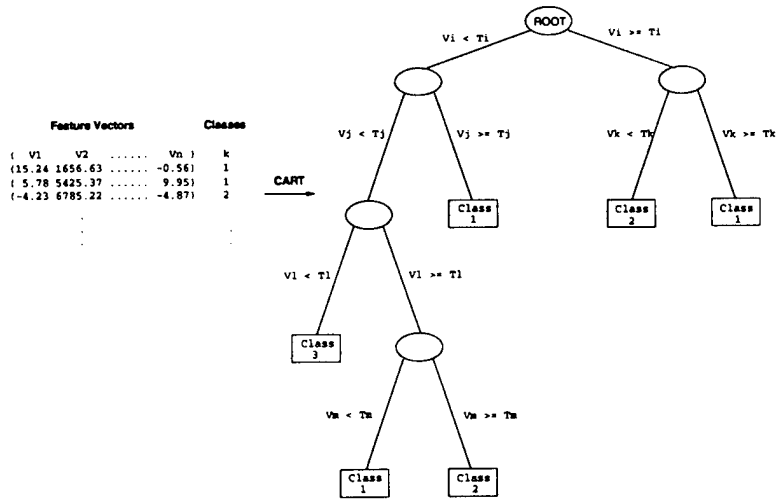
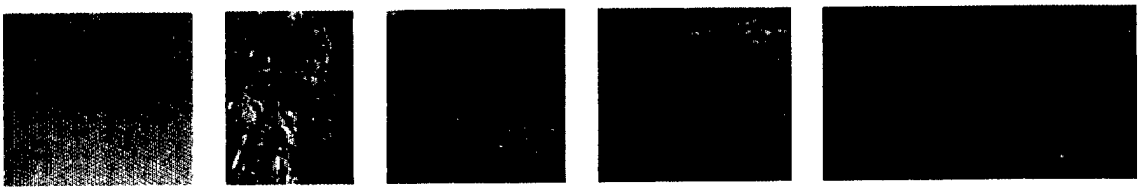
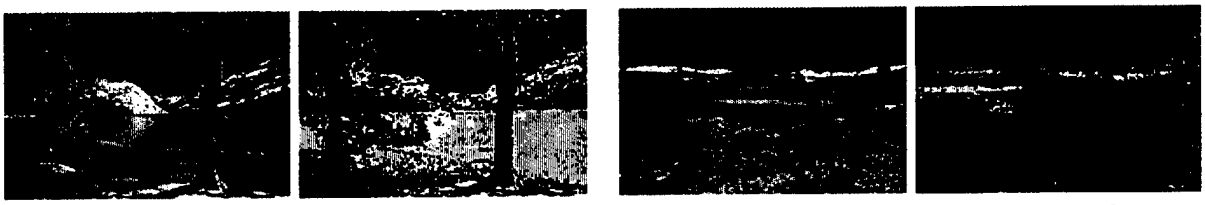


Figure 8: **Generating a classification tree using the CART algorithm.**



sky *stone* *river/lake* *grass* *tree/forest*

Figure 9: **Training color images used for creating the semantic overlay.**



color image *semantic overlay* *color image* *semantic overlay*

Figure 10: **Semantic analysis of outdoor scenes using the classification and regression trees (CART) algorithm.** No post-processing is performed. Color scheme: Deep blue for sky, yellow for stone, light blue for river/lake, light green for grass, deep green for tree/forest.

It takes about one minute on a Pentium PC to create the classification tree structure. After the classification tree is created, it takes only a few seconds to classify a given image to create the semantic overlay for a color image of 768×512 pixels. Figure 10 and 11 show the classification results on color and gray-scale images[2]. Each of the five different classes is given a unique "pseudo" color in the final result. The classification results are satisfactory for our application.

gray-scale image *semantic overlay* *gray-scale image* *semantic overlay*

Figure 11: **Semantic analysis of outdoor scenes using the classification and regression trees (CART) algorithm.** No post-processing is performed. Color scheme: Deep blue for sky, light blue for river/lake, light green for grass, deep green for tree/forest, white for non-classified regions.

For stereo matching purposes, we exclude regions classified as sky and water because feature-based matching in these regions is not reliable. We can obtain dense stereo matching in the water region by interpolation based on more reliable stereo matches bounding such regions.

SRI program : 279 matches, 1 error *INRIA's algorithm : 35 matches, 2 errors*

Figure 12: **Matching result using our program vs. INRIA's `image-matching` algorithm.** Dark points are the matches found. Lines shown are the disparity vectors. Our system found 279 matches including 1 mismatch (marked with white lines). INRIA's system found 35 matches including 2 mismatches.

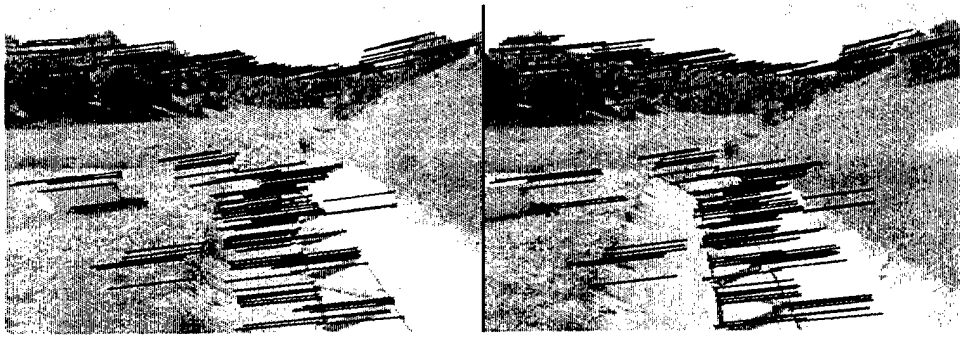
4.5 Geometric and Appearance Based Filtering

We next compute the fundamental matrix [12] that models the imaging geometry between the two images of the stereo pair and eliminate all conjugate pairs that fail to satisfy the epi-polar "rigidity" condition. Since we nominally assume that we know the internal camera parameters (as needed to fully exploit the semantic overlay), in an ideal system we would replace the epi-polar constraint with the more comprehensive collinearity constraint [13] to do the rigidity checking. We then further filter the surviving pairs on the basis of additional constraints derived from assumptions about scene geometry affecting two or more conjugate pairs.

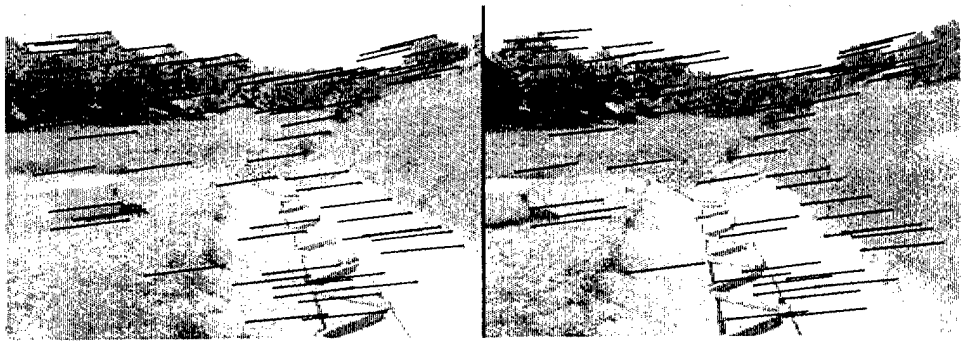
4.6 Occlusion Boundary Filtering

In this step, we eliminate matches on possible occlusion boundaries to avoid matching "psuedo features" composed of both foreground and background components. The procedure is based on the assumption that

[2]The original color figures can be accessed through the WWW at:
`http://www-db.stanford.edu/~wangz/project/stereo/J00/`



SRI program : 300 matches, 0 errors



INRIA's algorithm : 80 matches, 0 errors

Figure 13: **Matching result using our program vs. INRIA's image-matching algorithm.** Dark points are the matches found. Lines shown are the disparity vectors.

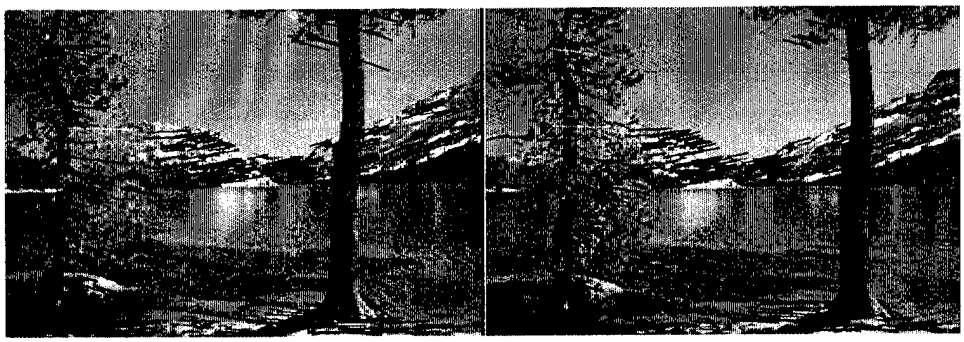
the intensity differences between corresponding points surrounding a given point on an occlusion boundary is less random than the differences surrounding a given point on a continuous surface.

Assume the match $\mathcal{I}_1(i_1, j_1)$ and $\mathcal{I}_2(i_2, j_2)$ is to be checked. The procedure is as follows:

1. As indicated in Figure 16, we partition the surrounding 16×16 pixel block of a match point into 4×4 sub-blocks.
2. Threshold the 16×16 block of intensity value differences between the first image and the second image to obtain a binary "difference block", denoted as B_0. The threshold is determined adaptively for each difference block to maintain about fifty percent ones in the 16×16 block. The threshold is typically around 12 for an 8-bit image.
3. Denote the 4×4 sub-blocks as $B_{i,j}$, where $i = 1, 2, 3, 4$ and $j = 1, 2, 3, 4$. Let
$$p_{i,j} = \frac{summation\ within\ B_{i,j}}{summation\ within\ B_0}\ .$$
4. If $\chi^2 = \sum_{i,j} \frac{(p_{i,j} - \frac{1}{16})^2}{\frac{1}{16}}$ is larger than a threshold 0.15, we discard the match pair.

Here we use the equivalent of a Chi-square test to evaluate the hypothesis that the intensity differences in the 16×16 block are uniformly distributed. The two thresholds currently used were determined based on experiments on real data. If the first threshold was chosen to produce 50% ones, then the Chi-square test with 15 degrees of freedom and a rejection threshold of $0.15 \times 128 = 18.2$ would result in a 25% probability of rejecting a valid match.

Figure 15 shows the performance of the occlusion boundary filter. We inserted an artificial occlusion boundary in each member of a pair of ground-level images. INRIA's program asserted some incorrect matches on the occlusion boundary. Our program eliminated all potential matches on the inserted occlusion boundary.



SRI program : 289 matches, 4 errors

INRIA's algorithm : 50 matches, 10 errors within the lake

Figure 14: **Matching result using our program vs. INRIA's `image-matching` algorithm.** Dark points are the matches found. Lines shown are the disparity vectors.

4.7 Mahalanobis Distance Filtering

For each semantic label (separately) we use the surviving pairs and their wavelet-based feature vectors to compute a 7×7 covariance matrix and then rank the remaining pairs on the basis of the Mahalanobis distance between members of the each conjugate pair. Based on the assumption that the differences between the wavelet characterizations of the members of a correctly associated conjugate pair can be approximated by a Gaussian process, we could set a threshold based on the Chi-squared distribution that allows us to eliminate any matches that have a probability of greater than (approximately) 2% of being in error. (The squared Mahalanobis distance has a Chi-squared distribution under the Gaussian assumption.) What if the Gaussian assumption does not hold?? We have found that the Mahalanobis distance consistently produces an acceptable ordering of the image points with respect to uniqueness for the class of natural scenes we are concerned with and it is possible to select a fixed threshold that virtually eliminates all but a very small percentage of errors – experimentally found to be less than 2 percent – while still returning on the order of 1-2 "certified correct" points per scan-line.

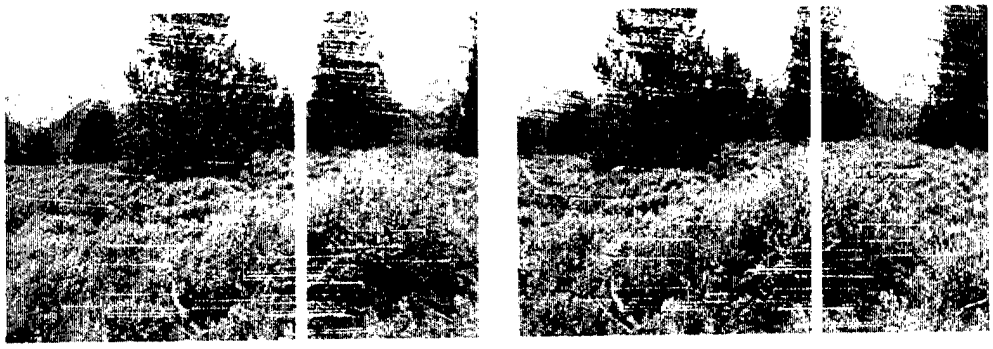
4.8 Dense-Modeling/Interpolation

The semantic overlay, certified correct matches, and computed epi-polar geometry, allow us to partition the images into *subregions* which are recursively processed by the above strategy.

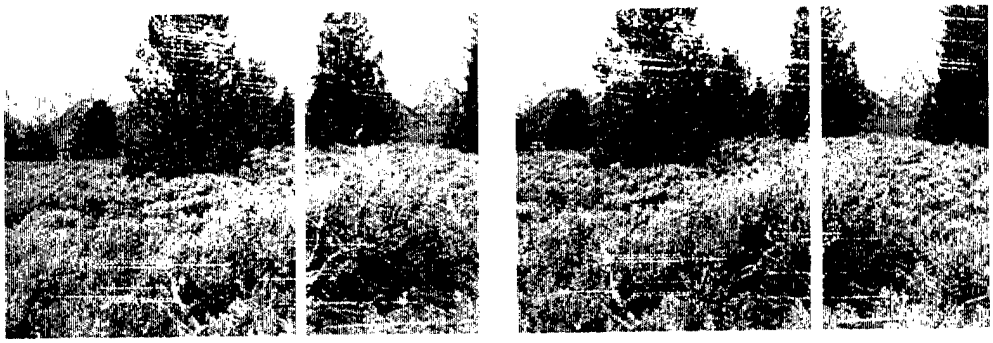
The recursive search in this step is limited to the set of pixels surrounding the corresponding epi-polar lines in each of the two images. Figure 18 illustrates the pixels to be examined in this step.

Since each subregion has fewer points to cause mismatches, we obtain additional (nominally) correct matches and thus the final number of conjugate pairs use to construct the 3-D scene model, while a function of scene content (e.g., the extent of the sky region), is not constrained by the size of our initial set of *certified conjugate points*. Dense modeling of depth is based on the assured correct matches and the semantic overlay to provide an informed basis for interpolation.

INRIA's algorithm : 90 matches, 10 errors (*)



SRI's algorithm : 194 matches, 0 errors



106 matches eliminated by the occlusion boundary filter (**)

Figure 15: **Performance of the occlusion boundary filter.** (*) A match near but outside the boundary of the inserted artificial foreground object is considered an error. (**) Of the 106 matches eliminated, 10 matches are near but outside the boundary of the inserted artificial foreground object.

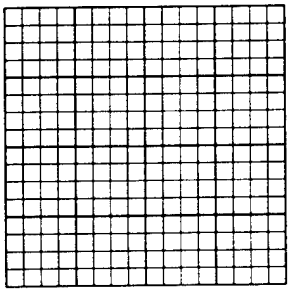


Figure 16: **Partition the** 16×16 **pixel block surrounding a match point into** 4×4 **sub-blocks for the occlusion boundary filtering.**

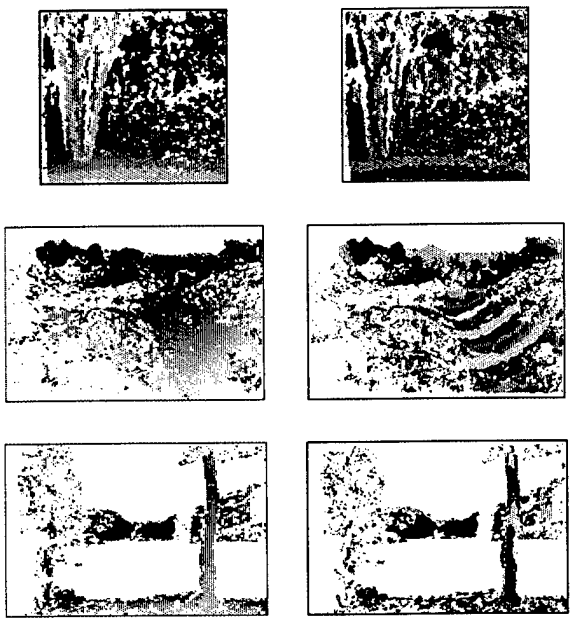


Figure 17: **Results from the recursive dense depth recovering process using our program.** The disparity image is shown. White regions are the no-match regions. No interpolation has been performed.

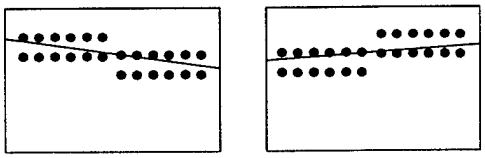
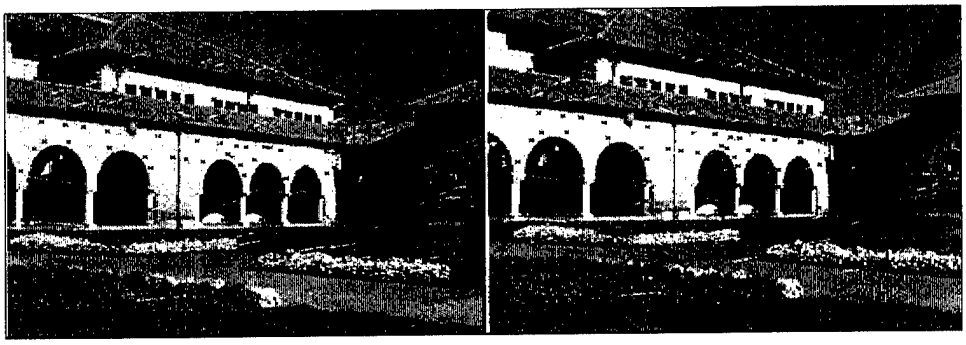


Figure 18: **Pixels surrounding the corresponding epi-polar line in each of the images are searched for final dense matching.**

At present, we have focused on sky and water constraints in exploiting the semantic overlay. Obviously, the sky regions are not assigned any finite depth value – they serve mainly to prevent the formation of incorrect correspondences or interpolation. It can easily be shown that for the imaging configuration we are assuming (known internal camera parameters and horizontal principal ray, we can estimate the elevation of a horizontal surface (e.g. a lake) relative to the focal point of the camera from a single correct correspondence of a point on or adjacent to the horizontal lake surface; and the distance to any point on the surface or surface-level boundary of the lake can then also be directly computed without any additional correspondences.

5 Experimental Results

SRI program : 296 matches, 0 errors



INRIA's algorithm : 50 matches, 10 errors

Figure 19: **Matching result using our program vs. INRIA's `image-matching` algorithm.** Dark points are the matches found. Lines shown are the disparity vectors.

The system has been implemented using C on a Pentium III LINIX PC. Figures 12, 13, 17, 14, 19 and 21 show sample matching results obtained using our system compared with using INRIA's `image-matching` algorithm.

We have performed a series of more than 100 experiments. For the ground-level natural outdoor scenes, we visualize the disparity maps to determine obvious errors (i.e., matches at least a few pixels away from the true match). For the aerial imagery, we use the SCT (self-consistency test) system [11] developed by SRI. Table 1 summarizes the data sets we have used. Figure 20 shows the cumulative distribution functions of errors, obtained from the SCT system. Our system was not intended to provide sub-pixel accuracy matching data.

6 Discussion

What conclusions can we draw from the experiments? Both the SRI and INRIA algorithms made almost no errors in the "Lambertian regions" of the three test scenes, but the filtering efficiency (retention of

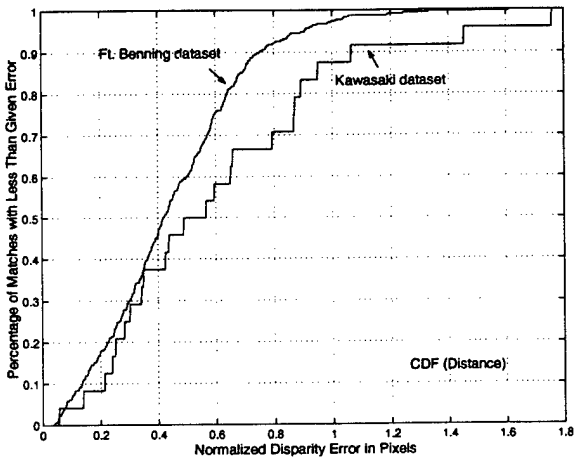


Figure 20: **Self-consistency test (SCT) results using the SRI program.**

SRI program : 273 matches, 2 errors

INRIA's algorithm : 90 matches, 1 error

Figure 21: **Matching result using our program vs. INRIA's `image-matching` algorithm.** Dark points are the matches found. Lines shown are the disparity vectors.

Location	Stanford Main Quad	Stanford Hill	Stanford Trees	Yosemite Lake	Ft. Benning	Kawasaki
# of Images	2	2	2	2	18 × 6 (**)	6 × 6 (**)
Image Dimensions (*rows* × *cols*)	294 × 445	512 × 768	233 × 256	294 × 447	400 × 400	450 × 450
Externally Supplied Data	F	F	F	F, S	F	F
Acquisition Geometry						
Type	ground	ground	ground	ground	air	air
Nominal Camera-baseline	2m	2m	1m	3m	N/A	N/A
Nominal Range in Disparity	15-60 pixels	60-100 pixels	0-20 pixels	20-50 pixels	0-30 pixels	0-30 pixels
Ground Surface Resolution (meters/pixel)	N/A	N/A	N/A	N/A	0.30	0.50
Performance of the **SRI** System						
# of Conjugate Pairs Returned	295	300	285	289	240-300 (*)	240-280 (*)
Valid Matches	> 99%	> 99%	> 99%	> 99%	> 99%	> 98%
Gross Errors	< 1%	< 1%	< 1%	< 1%	< 1%	< 2%
Self Consistency	N/A	N/A	N/A	N/A	100% within 2 pixels	100% within 2 pixels
Performance of the **INRIA** System						
# of Conjugate Pairs Returned	50	80	50	50	80	70
Valid Matches	> 80%	> 99%	> 90%	> 80%	> 99%	> 98%
Gross Errors	< 20%	< 1%	< 10%	< 20%	< 1%	< 2%

Table 1: **Performance comparison of the SRI system and the INRIA system.** F: fundamental matrices. S: semantic overlay computed. (*) The SRI system did not find more than 100 matches in some cases. (**) number of sets × number of images in a set. Lenses with 50mm focal length and 35mm film format were used for ground level scenes.

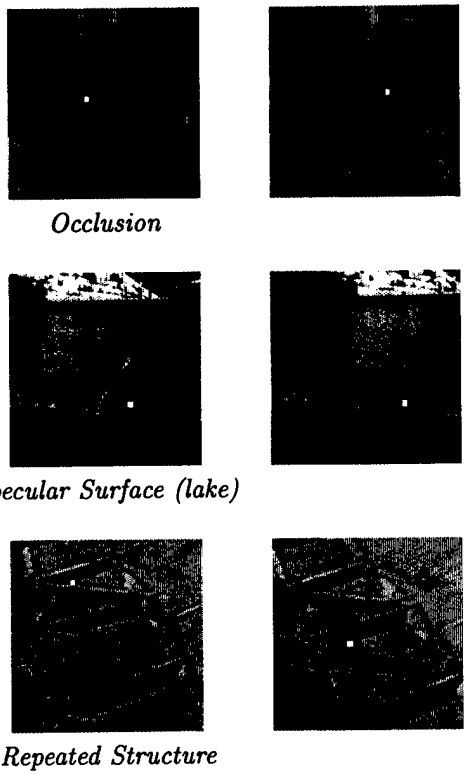


Figure 22: **Typical mismatches that we are trying to eliminate.** Bright points are the mismatches eliminated by our filters.

correct correspondences for an essentially zero error rate) was much higher for the SRI algorithm. Thus our ability to create a complete and valid depth model, even for the "normal" regions of the natural scenes, was significantly greater. In the case of the sky regions, both algorithms did well, but for the water there were significant differences. Here, as expected, without the semantic overlay, the INRIA algorithm had a high error rate – on the order of 20 percent of the returned matches.

We believe that the key to high efficiency in the filtering step is to have an initial collection of error-free matches to be used to construct the covariance matrix and thus also the rank ordering of the points with respect to expectation of a correct match. To the extent that incorrect correspondences are included, the correctness ordering (Mahalanobis distance) is "noisy" and a threshold chosen to eliminate almost all errors will be forced to also eliminate many correct correspondences. Thus, by preventing the sky and water regions from producing any correspondences, we improve the efficiency of the filters, even for parts of the scene outside of the sky and water regions. This explains why we were willing to pay a high computational price for the uniqueness computation in addition to the construction of the semantic overlay.

The uniqueness ranking that we assign to each conjugate pair is based on "all" the information present in both images. We assume that an ambiguity condition detected far from the original point in the containing image, or far from the associated epi-polar line in the conjugate image, still suggests an increased probability of an undetected mismatch (e.g., due to occlusion so that only one close but incorrect match is found on the correct epi-polar line itself) – we have found many examples where this is indeed the case (see Figure 22).

We assume that the only valid basis for certainty judgments is the consensus of informed independent opinions. Photometric measures based on different characterizations of the image intensity pattern are not likely to be truly independent. Constraints from the nature of the imaging process add additional necessary, but not sufficient criteria for a correct match. Thus, the other available information sources, especially constraints based on semantics, physical laws, and known or assumed scene geometry must be invoked if we are to have any hope of duplicating the performance of human stereopsis.

Stereo modeling of a natural scene requires a parallel (primitive) semantic overlay to provide a basis

for informed interpolation. This observation and its implications are central to our approach and a major departure from related work on this subject.

7 Conclusions and Future Work

In this paper, we addressed two related problems. First, we have explored the question of the extent to which judgments of similarity/identity can be made essentially "error-free." Most current approaches to robust matching focus on obtaining a consistent geometric model under a highly simplified set of assumptions about the imaging process and world being modeled. In the natural outdoor world, consistency is not sufficient; even a valid match does not insure correct depth recovery (e.g., the Tenaya-lake example). In the two image case, camera geometry constraints can, at best, restrict matching to epi-polar lines; at this point conventional systems usually rely on some form of local appearance matching and statistical arguments to complete the construction of the depth model. We show examples from non-contrived images where the statistics are valid but the matching is still incorrect. We argue that the HVS does not make these mistakes because it uses scene semantics as an additional, and more powerful, constraint on potential matches.

Second, we have examined the requirements for "human-level" stereo modeling in the natural outdoor world. Avoiding matching errors is only half the job: we can eliminate all the errors by eliminating all the matches. Consistency and statistical decision theory are not a sufficient basis for obtaining a relatively complete model when a significant portion of the scene content is "unmatchable" (i.e., when such matching is based strictly on intensity variations in the imagery). Interpolation into the unmatched regions can only be accomplished in a principled way if a semantic constraints are invoked and if semantic modeling accompanies geometric recovery.

Since the matching problem is "open-ended," this paper is still *work-in-progress*. We are attempting to better define the requirements of the semantic overlay, to make its automatic construction more robust, and to use it more effectively in the stereo matching process.

Acknowledgments

We would like to thank Yvan Leclerc, Quang-Tuan Luong, Marsha Jo Hannah, and various other researchers of the SRI AI Center for help. In particular, Yvan was involved in the early work leading to this paper and made important contributions with respect to our approach to evaluating uniqueness. We would also like to thank INRIA for making publicly available a highly robust image matching program for purposes of comparative evaluation.

References

[1] E B Barrett, P Payton, M H Brill and N N Haag, Linear Resection, Intersection, and Perspective-independent Model Matching in Photogrammetry: Theory, *Appl. Digital Image Processing XIV*, editor A. Tescher, Proc SPIE 1567, p. 142-169, 1991.

[2] L Breiman, J H Friedman, R A Olshen and C J Stone, *Classification and Regression Trees*, Chapman and Hall, 1984.

[3] I Daubechies, Orthonormal bases of compactly supported wavelets, *Communications on Pure and Applied Mathematics*, 41(7):909-996, October 1988.

[4] I Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, 1992.

[5] M A Fischler and R C Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *CACM*, 24(6):381-95, June 1981; also, *Readings in Computer Vision*, (M A Fischler and O Firschein, eds.), Morgan Kaufmann, pp 726-40, 1987.

[6] M A Fischler, Robotic Vision: Sketching Natural Scenes, *ARPA Image Understanding Workshop*, Feb 1996.

[7] P Fua and Y G Leclerc, Registration Without Correspondences, *CVPR* Seattle, June 1994.

[8] *B Julesz, Foundations of Cyclopean Perception*, Univ. of Chicago, Ill, 1971.

[9] G Kaiser, *A Friendly Guide to Wavelets*, Birkhauser, Boston, 1994.

[10] Y G Leclerc, Q.-T. Luong and Pascal Fua, Self-consistency: a Novel Approach to Characterizing the Accuracy and Reliability of Point Correspondence Algorithms, *DARPA Image Understanding Workshop*, 1998.

[11] Y G Leclerc, Q.-T. Luong and Pascal Fua, Characterizing the Performance of Multiple-image Point-correspondence Algorithms using Self-consistency, *Proceedings of the Vision Algorithms: Theory and Practice Workshop (ICCV99)*, Greece, September 1999.

[12] Q.-T. Luong and O D Faugeras, The Fundamental Matrix: Theory, Algorithms, and Stability Analysis *Int J of Computer Vision*, 17(1):43–76, 1996.

[13] D P McReynolds and D G Lowe, Rigidity Checking of 3D Point Correspondences Under Perspective Projection, *IEEE PAMI*, 18(12):1174-85, Dec 1996.

[14] Y Meyer, *Wavelets: Algorithms & Applications*, SIAM, Philadelphia, 1993.

[15] E M Mikhail, *Observations and Least Squares*, IEP, New York, 1976. Also publ. by Harper and Row, 1980.

[16] J Mundy and A Zisserman (eds), *Geometric Invariance in Computer Vision*, MIT Press, Cambridge, Mass., 1992.

[17] X Pennec, J.-P. Thirion, A Framework for Uncertainty and Validation of 3-D Registration Methods Based on Points and Frames, *Int J of Computer Vision*, 25(3):203-229, Kluwer, 1997.

[18] P J Rousseeuw, *Robust Regression and Outlier Detection*, Wiley, New York, 1987.

[19] S B Stevenson and C M Schor, et al., Human Stereo Matching is Not Restricted to Epipolar Lines, *Vision Research*, Elsevier, 37(19):2717-23, Oct 1997.

[20] P H S Torr, Motion Segmentation and Outlier Detection *University of Oxford Thesis*, 1995.

[21] P H S Torr and D W Murray, The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix, *Int J Computer Vision*, Kluwer, 24(3):271-300, Sept/Oct 1997.

[22] J Z Wang, G Wiederhold, O Firschein and X W Sha, Content-based Image Indexing and Searching Using Daubechies' Wavelets, *Int J Digital Libraries(IJODL)*, 1(4):311-328, Springer-Verlag, 1998.

[23] J Z Wang, J Li, G Wiederhold and O Firschein, System for Classifying Objectionable Images, to appear in *Computer Communications J*, Elsevier Science, 1998.

[24] J Weng, T S Huang, and N Ahuja, Motion and Structure from Two Prospective Views: Algorithms, Error Analysis, and Error Estimation, *IEEE PAMI*, 11:451-476, 1989.

[25] Z Zhang, R Deriche, O Faugeras, Q.-T. Luong, A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry, *Artificial Intelligence*, 78(1-2):87-119, Elsevier, Oct 1995.

[26] Z Zhang, et al., Determining the Epipolar Geometry and Its Uncertainty: a Review, *Int J Computer Vision*, 27(2):161-95, Kluwer, 1998.

[27] Special Issue on Wavelets and Signal Processing, *IEEE Trans. Signal Processing*, Vol.41, Dec. 1993.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
08-2000	Final	

4. TITLE AND SUBTITLE
Representation, Modeling, and Recognition of Outdoor Scenes

5a. CONTRACT NUMBER
DACA76-92-C-0008

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Martin A. Fischler and Robert C. Bolles

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025-3493

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 N. Fairfax Drive
Arlington, VA 22203

U.S. Army Topographic Engineering Center
7701 Telegraph Road
Alexandria, VA 22315-3864

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
ERDC/TEC CR-00-4

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

14. ABSTRACT

The goal of this project is to advance the state-of-the-art in scene interpretation for autonomous systems that operate in natural terrain. In particular, techniques are being developed for representing knowledge about complex cultural and natural environments so that a computer vision system can successfully plan, navigate, recognize, and manipulate objects, and answer questions or make decisions relevant to this knowledge. The results to date include the development of new representations and techniques for rapidly modeling terrain from multiple images, and for the recognition and reliable labeling of such scene attributes and components as color, texture, shadows, and a variety of linear structures (skyline, ridgelines, road, etc.). The most recent results are detailed in three papers included as appendices to this report.

15. SUBJECT TERMS
machine vision, automated scene analysis, object recognition, terrain modeling, linear delineation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Robert C. Bolles
Unclassified	Unclassified	Unclassified	SAR	84	19b. TELEPHONE NUMBER (include area code) 650-859-4620