

SRI International

Artificial Intelligence Center

Annual Progress Report

August 3, 1998

An Integrated Feasibility Demonstration for Automatic Population of Geospatial Databases

SRI Project Number:

ECU-1515

Contract Number:

NMA202-97-C-1004

DARPA Order Number:

E645

Prepared by:

Aaron J. Heller, Sr. Computer Scientist
Martin A. Fischler, Principal Scientist
Robert C. Bolles, Program Director
Chris I. Connolly, Computer Scientist
Robert Wilson (Vexcel, Inc.)
James J. Pearson (GDE Systems, Inc.)

Prepared for:

Michele Motsko, Physical Scientist
Applied Research Branch, MS D-84
National Imagery and Mapping Agency
4600 Sangamore Rd.
Bethesda, Maryland 20816-5003

Approved by:

C. Raymond Perrault, Director

CONTENTS

1	Introduction	5
2	Technical Background	5
2.1	The Core Problems in Designing an Automated Recognition System	6
3	An Architecture for Feature Extraction	7
4	Road-Extraction Techniques	10
4.1	State of the Art	10
4.2	Road-Extraction Architecture	11
4.2.1	Low-Resolution Analysis	13
4.2.2	High-Resolution Analysis	15
4.2.3	Interactive Road Editor	17
4.3	Evaluation Rationale, Metrics, and Procedures	17
4.3.1	The Evaluation Process	20
4.4	Progress in Road Modeling	21
4.5	Virtual Laboratory Facilities for Road-modeling	22
5	Building-Extraction Techniques	23
5.1	Cue Points and Building Models	24
5.2	Cue Point and Initial Model Generation	25
5.3	Building Extraction, Given a Cue Point and Building Model	28
6	Current Status and Future Plans	30
6.1	Status	30
6.1.1	Roads	30
6.1.2	Buildings	31
6.2	The Year2 Feasibility Demonstration, January 1999	31
6.3	The Year2 High-Level APGD Work Plan	32
A	Specification for APGD datasets	36

B	Sensor Geometric Models	38
B.1	Central Perspective Camera	39
B.2	Orthographic Projection	39
B.3	Fast Block Interpolation Projection	39
B.4	Rational Polynimial (RPC)	41
B.5	Composite	41
C	Interoperation Specification	43
C.1	Scope	43
C.2	SocetSet to CME File Hierarchy	43
C.3	Project Description File	43
C.4	Example Project Description File	44
C.5	Pyramid Descriptor File	45
C.6	Example Pyramid Descriptor File	46
C.7	Image Properties File	46
C.8	Sensor Geometric Models ¹	46
C.9	“TEC Header” Files	46
C.10	RPC Files	46
C.11	Example RPC File:	47
C.12	CAMGRID Files	47
C.13	Example CAMGRID File:	48
D	Proposed APGD Evaluation Procedures	50
D.1	Report Format	52
D.2	The Road Evaluation Process	53
D.2.1	Segment Geometry	53
D.2.2	Segment Attributes	54
D.2.3	Network Topology	54
D.3	The Building Evaluation Process	54
D.4	APGD Evaluation Philosophy and Rationale	55

¹a.k.a. “Camera Models”

D.4.1	Discussion of Critical Issues and Assumptions	56
E	APGD Evaluation Data Formats	61
E.1	Introduction	61
E.2	Syntax and File Format	61
E.2.1	Tag	61
E.2.2	Attributes	61
E.2.3	Images	61
E.2.4	Objects	62
E.3	Primitives	62
E.3.1	Object Space Coordinates	62
E.3.2	Image Plane Coordinates	62
E.3.3	Points	63
E.4	Road Network	63
E.5	Buildings	63
E.6	Complete Example	64
F	Videotape of First Annual Demonstration	68

Abstract

In this Automatic Population of Geospatial Databases (APGD) report, we describe our progress in developing and evaluating a generic context-based architecture (which we call the BOS for the Battlespace Observer System) and algorithmic techniques to radically reduce the human effort required to extract cartographic features, especially roads and buildings, from aerial and satellite imagery. We also describe progress in supporting the DARPA APGD community by constructing verified and well-documented datasets and evaluation procedures to allow interested researchers to experimentally evaluate their extraction techniques in a common framework. Our APGD Community oriented activities are accessible at SRI's APGD web site, <http://www.ai.sri.com/~apgd>, and the APGD "Virtual Laboratory," <http://www.ai.sri.com/~apgd/vl>.

In a formal April 1998 presentation, we successfully demonstrated the "end-to-end" extraction of all the roads and buildings at the Ft. Benning McKenna Military Operations in Urban Terrain (MOUT) facility and surrounding area, duplicating a professionally produced model with less than one-tenth the required human effort. We describe the technical and methodological advances required to achieve this benchmark goal and discuss our future plans.

1 INTRODUCTION

The goal of the Automatic Population of Geospatial Databases (APGD) project within the Image Understanding for Battlefield Awareness (IUBA) Program is to develop, demonstrate, and evaluate technology for rapidly and robustly extracting three-dimensional (3-D) cartographic features, such as roads and buildings, from aerial and satellite imagery. The uses for high-resolution, geospecific 3-D site models are rapidly increasing – fueled by dramatic advances in computer graphics and simulation. The volume, number of sensor modalities, and resolution of data suitable for constructing site models are also dramatically increasing. The missing piece is the technology for creating 3-D models without the need for significant human interaction. The purpose of the APGD program is to develop these techniques. Under this program, SRI, and its team members GDE and Vexcel, are developing procedures for radically reducing the need for human intervention in the APGD extraction process.

Our primary program goal is to reduce the amount of human interaction time by a factor of 10 to 100. We are concentrating on human interaction time, because computational resources for the automated parts of the task continue to significantly increase in capability and decrease in cost. Our high-level approach is to use context, such as terrain geometry and land-cover classification derived from imagery, general site knowledge (for example, rural or urban), previous maps, and construction practices, to guide and constrain the extraction process.

In the remainder of this report (which includes a 10 minute videotape) 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 unsolved in general. We have made significant progress in automatic techniques for recovering scene geometry and can automatically recognize objects that have explicit geometric descriptions. However, except for the case where some 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 an object that can be found at a known geographic location, such as a submarine in a specific “submarine pen,” then we can usually determine if the object (in this case the submarine) is or is not present. 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 visual sensing 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 is the requirement to express a typically function-oriented description of the object of interest, such as a road, 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., all of the 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 provide all of information needed to establish if the definition is (or is not) satisfied.

Thus, the first problem to be solved is to provide a computational redefinition of the 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 thus 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 road extraction technique 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 Figure 3 and Figure 4). Another, more practical, measure taken to control the computational requirement is to restrict all algorithms to be $O(n \log n)$ or $O(n)$ in their theoretical complexity.

3. Self-Evaluation (i.e., 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 an algorithm that scores very well in a given benchmark evaluation, but for which the designer is unable to provide performance characterizations 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.

3 AN ARCHITECTURE FOR FEATURE EXTRACTION

It is our contention that the image understanding community has developed several feature-extraction algorithms that, when applied to an appropriately constrained problem and properly parameterized, can successfully perform their intended functions in difficult “real-world” situations. Therefore, we believe that development of an architecture that supports context-based application of these algorithms will significantly increase automation of feature-extraction tasks, such as road and building extraction.

In this project, we have developed an architecture of this type, called the BOS, or Battlespace Observer System (Figure 1). The BOS accepts task requests, such as “extract the roads in region X (specified in world coordinates),” collects contextual information about region X, analyzes the imagery covering region X, selects relevant algorithms and parameter settings, applies the algorithms, and then evaluates the results.

The BOS is implemented in a highly-integrated 3-D cartographic modeling environment (variously called the CME, RCDE, or 3DIUS), that runs on SGI and Sun Microsystems workstations.

In the BOS, the contextual analysis is performed in two phases. First, the feature-extraction managers (FEMs) select the site information and data that are relevant for the requested task. Second, the context-based algorithm control system (CBACS) selects and parameterizes algorithms to apply to the data selected by the FEM.

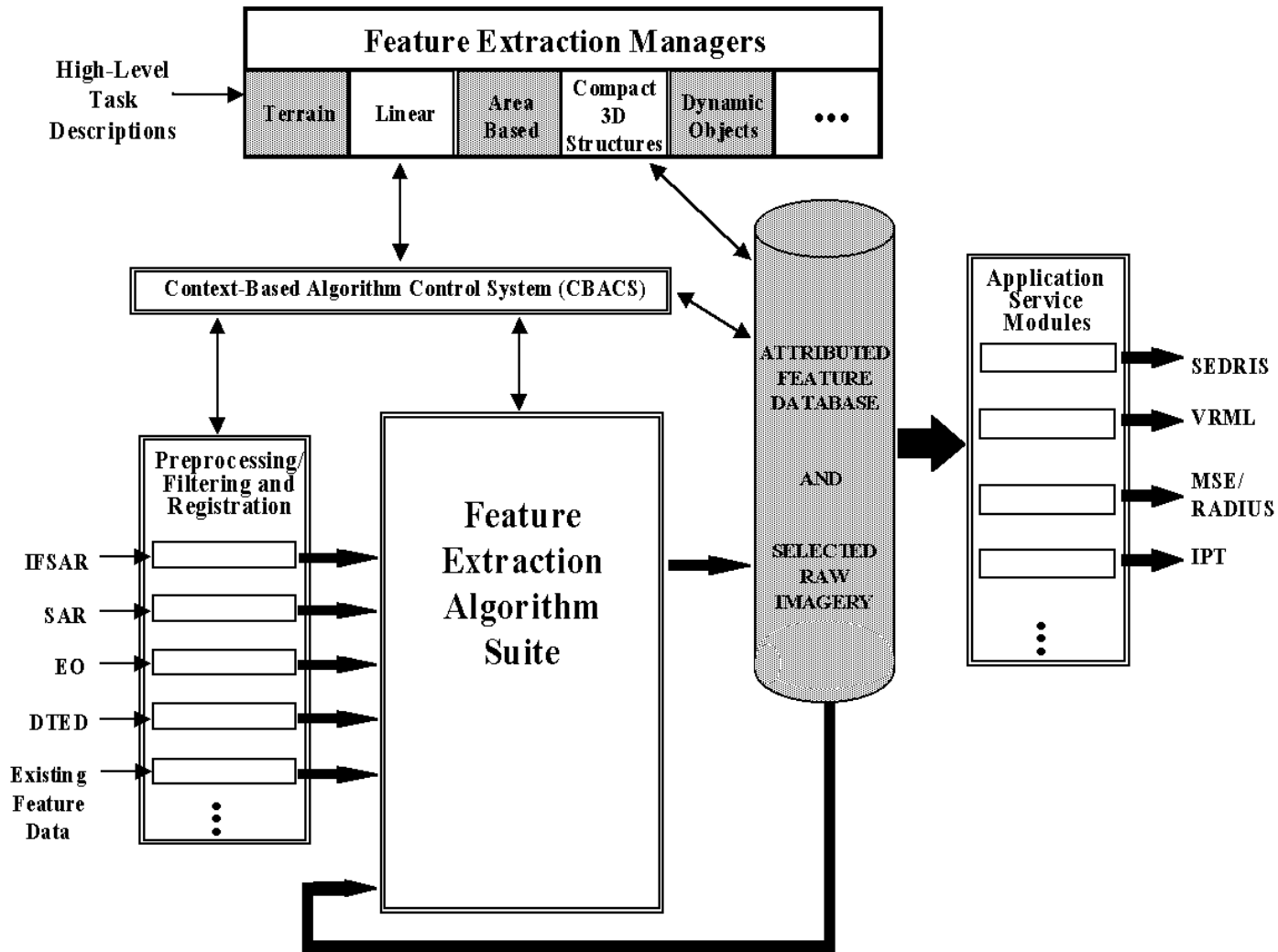


Figure 1: Block diagram of the Battlefield Observer System (BOS).

The FEM contains a Prolog engine that applies a set of rules to make inferences about the relevance of imagery and contextual information for a specific task. For example, if the task is to extract the railroads in a region, the FEM “knows” (i.e., it has rules stating) that, with appropriate acquisition geometry, railroads typically appear as high-contrast bright lines in SAR imagery. Therefore, if SAR imagery is available for the region, it packages up each SAR image and its metadata and sends them to CBACS as suggestions for data to use to extract railroads in the region of interest. We call these suggestions *context objects*² because they combine imagery and contextual information within one task-specific representation.

In the BOS, each algorithm is expected to have an associated description of the conditions under which it operates. For example, if an algorithm only works with black-and-white images having a GSD in the range of 1 to 5 meters, it is expected to have a precondition stating this constraint. We call these preconditions the *filter set* for a given algorithm. A filter set represents a range of contextual variables over which the algorithm is expected to be competent. Within each filter set, an algorithm has specific parameter settings that can be tuned, but are assumed to apply over the range described by the filter set. A given algorithm can be represented more than once in CBACS, especially if different filter sets require customized parameter settings to achieve optimal performance for the algorithm.

Given a list of suggestions (i.e., context objects), CBACS tries to find algorithms that are appropriate. In particular, for each context object, CBACS examines the list of available algorithms and associated parameter settings and filters out those that do not have their preconditions met by the information in the context object.

When CBACS finds an algorithm and parameters that are appropriate for a context object, it executes the algorithm with the parameter settings specified in the context object. Each context object represents information sources that an algorithm can use to perform its task. For extraction tasks, images provide the primary source of information, whereas for refinement, the feature to be refined and the images within which the feature is visible all serve as information sources. CBACS maintains a database of the parameters associated with an algorithm and the contexts that caused them to be selected so the selection criteria can be tuned over time to improve the performance of an algorithm and as a diagnostic aid if it becomes apparent that a given algorithm is being applied under inappropriate conditions.

The combination of FEMs and CBACS provide a modular approach to integrate feature-extraction techniques that work well in very narrow situations into a feature extraction and attribution system that is reliable over the broad range of real-world conditions. The FEMs and CBACS use context objects as a convenient representation for the information that IU algorithms require to perform extraction tasks. Given this representation, a new algorithm can be incorporated into CBACS by writing a “wrapper” that describes the parameter settings appropriate for different contexts.

For the first annual demonstration in April 1998, CBACS was used primarily within the interactive editing phase. It helped a user fill in missing road segments, adjust automatically extracted segments, and delete mistakes. In particular, CBACS provided three operations to the user: create a new road segment, extend an existing road segment, or link together two existing road segments.

²The word *object*, as used here, refers to object-oriented programming techniques, rather than *object features* in the world.

In each case, it prompted the user to select two points in the site to be joined by the corrected road segment. CBACS then automatically selected two or three alternative algorithms and parameter settings to perform the requested operation. The alternative results were presented to the user, who could then choose the best result for inclusion in the site model.

The BOS is currently being extended in four ways. First, the FEM has been expanded to include five basic task types: *extract*, *refine*, *annotate*, *verify*, and *interpolate*. The April 1998 demonstration of the interactive road editor depended solely on *interpolate* tasks. Rules have been encoded for selecting relevant context for these five task types. Second, the context-based selection of algorithms is being strengthened so that CBACS can select and tune algorithms more effectively based on the characteristics of the local area. Third, the results of the automatic extraction process are going to be annotated and then evaluated in terms of their local context. For example, after the road network has been automatically extracted, 3-D properties of the road segments, such as along-road slope and curvature, will be computed and stored. Then a region-based analysis, encoded as rules, will evaluate the plausibility of each extracted road in its local context. For example, in a mountainous region, a road can have a higher along-road slope and curvature than in a flatter area. Fourth, a global analysis of the network will be performed to identify possible gaps in the network that should be re-analyzed by the system, possibly with more computationally expensive techniques. These extensions will significantly increase the BOS's sensitivity to the local contextual information.

4 ROAD-EXTRACTION TECHNIQUES

4.1 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]). Although numerous algorithms have been developed to date, almost all the linear delineation algorithms are *trackers* in that they follow a single path. They generally require a start point, direction, and width information. The main distinction between these trackers is whether they depend on detail internal to the road. In high-resolution imagery, a linear feature, such as a road, appears as a ribbon with internal structure (e.g., painted lines, ruts), whereas at low resolution, it is seen as 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].

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.

Path-cost optimizers come in several 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 (e.g., a number inversely proportional to the likelihood that a pixel is actually a road or road-edge pixel) 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— i.e., 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 track the weak road since the tracker prefers to jump 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 few systems that attempt, in a single integrated operation,³ to extract complete road networks from aerial images without an externally supplied image-specific initialization or guiding sketch.

In the next few sections of this report, we briefly 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 2). Five distinct linear-delineation algorithms are invoked: (a) a graph-theoretic network-delineator to extract global road-network topology and initial centerline estimates from low-resolution synoptic imagery (e.g., an orthophoto of the complete site being modeled); (b) a pair of 3-D model-based optimization algorithms to refine the local road-geometry and produce a 3-D ribbon using multiple high-resolution images, and (c) a collection of interactive path-delineators, including both 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.

4.2 Road-Extraction Architecture

The sequence of operations performed by the current road-extraction process, shown in Figure 2, involves three distinct phases:

Low-Resolution Analysis. The low-resolution analysis – which we call LD (for linear delineation) – automatically extracts a road network from a single aerial image and then drops it to a terrain model to produce a 3-D description of the network. The resulting network is a topological description of the visible roads in terms of a road center-line model. This process is fully automatic and operates on images with a nominal ground resolution of two to eight meters per pixel. (The first five boxes in the processing sequence, shown in Figure 2, are part of the low-resolution analysis.)

High-Resolution Analysis. The high-resolution analysis (HRA) refines the road network produced by LD by projecting it into several high-resolution images, accurately positioning the road bed, and determining attributes such as road width and along-road gradient. This analysis is fully automatic. The output is an attributed road network model. The high-resolution images are assumed to have a ground sample distance of less than one meter. (The sixth and seventh boxes in the processing sequence, shown in Figure 2, are part of the high-resolution analysis.)

³Here we distinguish between the trackers discussed earlier, and a full network delineator. In theory, a single-path tracker can be applied repeatedly to an image to extract a complete road network, but the number of distinct paths through even a reasonably sized network can make this strategy computationally infeasible for many practical problems.

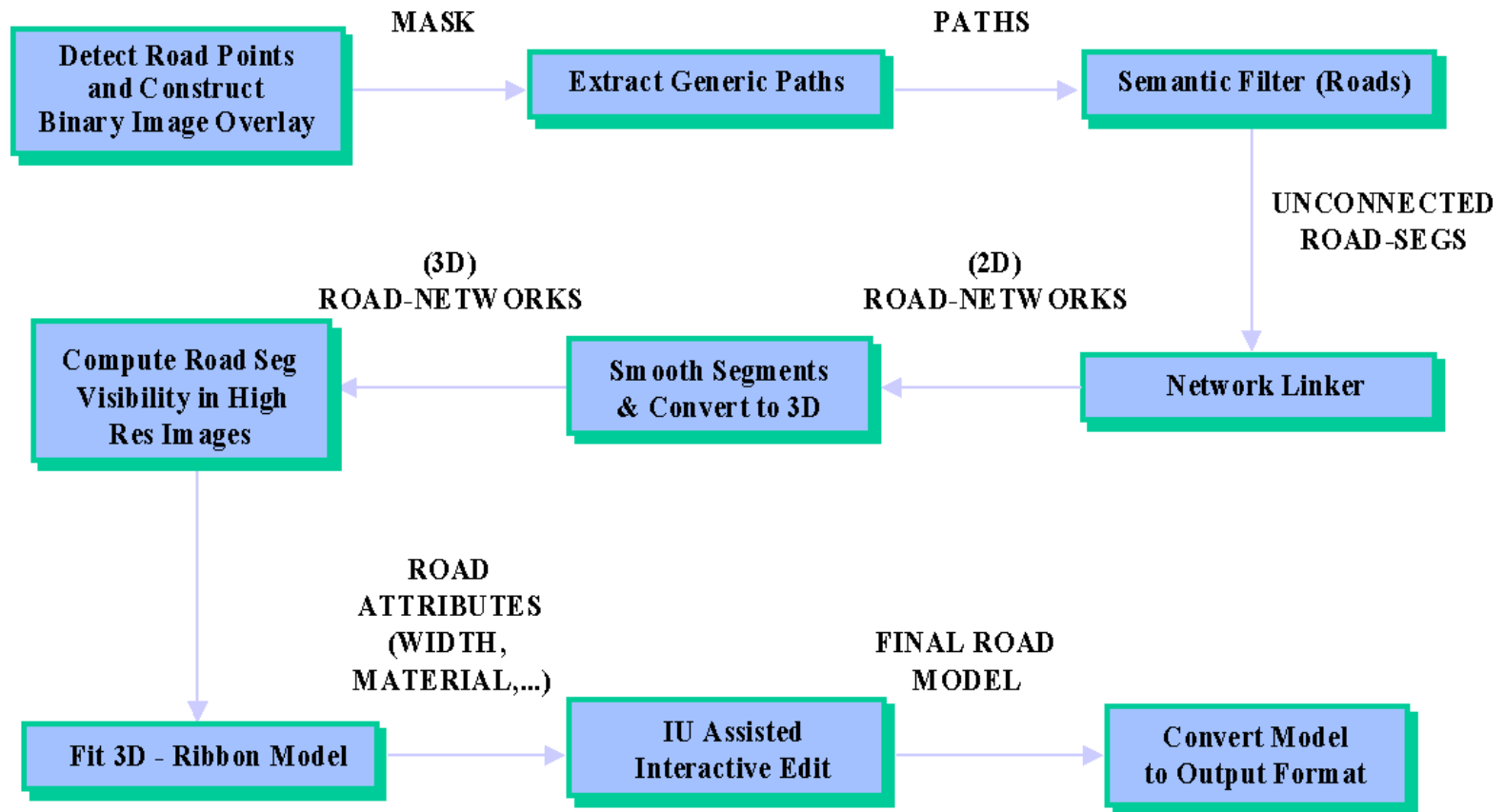


Figure 2: Process Flow Graph For Road-Model Construction.



Figure 3: Ft. Benning McKenna MOOT orthomosaic.

Interactive Road Editor. Given the results of the automatic extraction procedures, the final step is for a person to edit the extracted road network so that it meets the application specifications. To facilitate this step, we provide a highly-efficient interactive editor that is able to employ a significant inventory of automated tools under human control. (The eighth box in the processing sequence, shown in Figure 2, is the editing phase.)

These three phases are more fully described below. In addition, the road-extraction process is described in greater detail in a separate technical paper to be published in the 1998 IU Workshop proceedings.

4.2.1 Low-Resolution Analysis

Step 1. Detect potential road points in 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 Figure 3 and Figure 4).

This 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 appear similar to that of a small stretch of road. It is also the case that tunnels, trees, buildings, clouds, and so forth, can occlude sections of a road, causing an apparent break in its continuity.



Figure 4: Linear structure mask created from Ft. Benning orthomosaic.

Step 2. Assemble potential path-points into dense segments by using a fast minimum-spanning-tree (MST) algorithm.⁴ Recover the longest smooth segments (i.e., those 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 linear structure mask (Figure 4); the output is a list of disjoint segments called *RPATHS*, that represent potential roads (Figure 5).

The generic linking algorithm must solve several problems. Given that the detection task is not error free – 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 for which we are searching.

Because of possible errors in assumed connectivity, and because roads and other non-road linear structures can 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 be able to discriminate between road and non-road segments in the networks returned by the generic linker and permit some relinking of the network.

Step 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 and the generic parsing process.

⁴Although the MST does not actually ensure the densest connectivity, it usually provides a very good approximation to this condition.

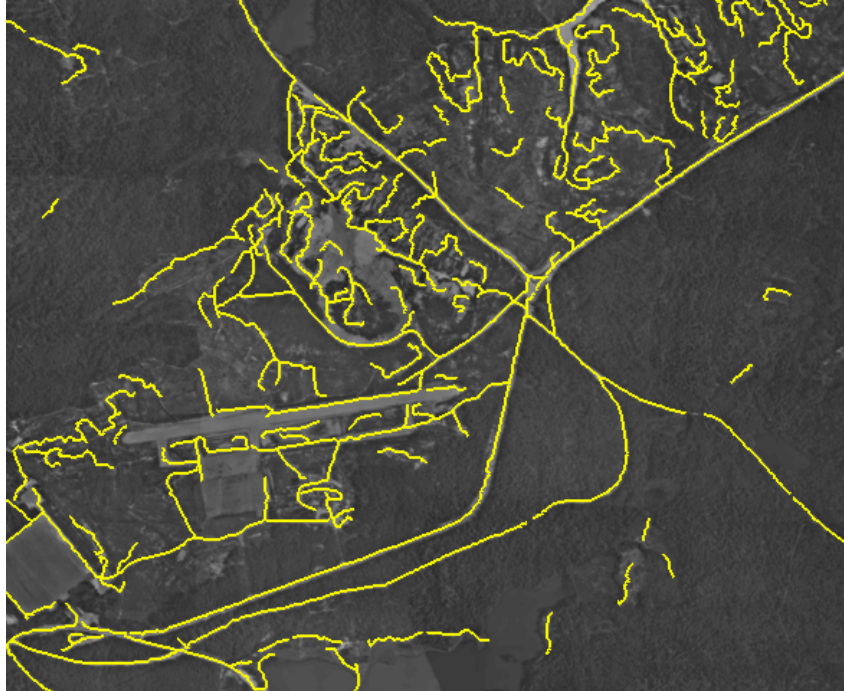


Figure 5: Disjoint segments, called *RPATHS*, that represent potential roads.

Step 4. Use a recently developed linking technique and representation schema [Fischler and Heller, 1998], capable of expressing perceptual and semantic constraints, to imply a network of paths that is very likely to include the road network to be modeled (Figure 6).

4.2.2 High-Resolution Analysis

As currently implemented, the result from the semantic linker phase is a 2-D network of eight-connected pixel chains that correspond to the road centerlines of 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 mono-plotting 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, we build an initial road segment visibility table that indicates in what images a given segment should be visible disregarding inter-object 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 implement in a very general way, so that any objects already modeled are checked.

In the case of the April demonstration using the Ft. Benning dataset, we are essentially in a



Figure 6: Final output from the low-resolution road-extraction phase.

cold-start mode and the only a priori spatial objects we have are the two terrain elevation models: 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 remove images from the entries in the road segment visibility, 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 (MBO) system on each of the road segments in the network, using all of the images that have passed the above tests.

This MBO system has been described in detail in other papers, but briefly, it attempts to align the two edges of the road with high gradients in the images while not deviating significantly from an a priori 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 over 5 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 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.

Experiments have shown that using more than 4 or 5 images for each road does not provide significant improvement and in some cases can degrade performance. Therefore it is important to 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.

4.2.3 Interactive Road Editor

The interactive road editor allows a user to view and correct mistakes in an existing road network model. The editor works in two phases. In the first phase, the editor visits potential “trouble spots” to allow the user to make changes. In the second phase, the editor scrolls through the road network, allowing the user to view the length of each road and make any other necessary corrections.

In the first phase, the interactive road editor estimates the *approximate degree* of each node (intersection) in the road network by counting the number of road segment endings that all occur within a certain radius of the node. A high approximate degree indicates a point where many road segment endings occur in close proximity, and hence, an area most likely to need human attention. Nodes in the network with an approximate degree four or greater are inserted into a priority queue and reviewed in order of approximate degree, so that any errors in these “high-payoff” nodes are addressed first.

In the second phase, a cost function is computed over the network, whose minimum is at the node with the highest approximate degree. The roads are then traversed by a gradient descent on the cost function, until all road segments in the network have been traversed. The use of a cost function allows relatively long chains of road segments to be traversed as a single road.

At any point in either phase of the interactive editing process, the user may make corrections to the road network by adding or deleting roads. The user can add roads by selecting a pair of points to be connected. CBACS selects and runs algorithms (with parameters) based on the context at those points. The user is presented with the results and can choose the best one for inclusion in the road network. When using the interactive road editor, the operator has access to the full suite of site-modeling and editing tools that were developed as part of the RADIUS project [Heller *et al.*, 1996, Fua, 1996], so that if none of the results offered by CBACS are satisfactory, the RADIUS tools can be used to create or delete features. Another alternative is to accept the closest result and interactively modify it as needed.

4.3 Evaluation Rationale, Metrics, and Procedures

This section briefly describes our evaluation process. A more complete description can be found in the appendices. Appendix D defines the metrics, presents an example, describes the form of an evaluation report, and discusses several issues and assumptions. Appendix E describes the evaluation data formats.

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., days to



Figure 7: Final output from the high-resolution road-extraction phase.

weeks), the cost of automation largely amounts to the time spent fixing the errors and shortcomings 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 road-extraction evaluation is to quantify the performance (in terms of human-interaction-time, correctness, completeness, and branching factor), of our overall system and its automated component, to recover a road-centerline-data-model from multiple images of a site.

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 becomes increasingly narrower until it finally disappears or 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 portions of the scene 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 Reference Road-Segments (RRSs) intersect. This information 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-usable-width, surface-material-type, etc.

Each component of the data-model (other than the don't-evaluate components) can be extracted separately and should be evaluated separately. Therefore, if some component is missing in the reference data, or can't be computed from the imagery, or has different implications in terms of its difficulty or importance in compiling a final model, we have the flexibility to evaluate the separately available components. A key element of the Road-Centerline Data-Model is the independent treatment and evaluation of width-type attributes, such as number-of-traffic-lanes, from the problem of establishing centerline geometry and topology. Width measurements are difficult (sometimes impossible) to make locally, while a human editor can insert a number-of-traffic-lanes estimate for miles of road-length with a single entry. In this document, we focus on the evaluation of methods for recovering the RRS portion of the Road-Centerline Data-Model.

A critical issue is the need for, and use of, tolerances. Our purpose (in the APGD program) is to eliminate most of the manual effort required to produce a cartographic product equivalent to that generated in current practice. We note that it would be impossible for a group of cartographers to

produce Road-Centerline Data-Models in which the RRSs are in perfect agreement with each other. This residual variation defines the acceptable tolerance for an automated process. In particular, *we must not assume that any deviation from a provided Reference Model is an error in the Derived Model*, but rather that the correct answer lies within some tolerance band around the RRS. If we do not make this provision, the Derived Model will always be assigned an evaluation score that is lower than a correct accounting would dictate. In cases where we cannot reasonably estimate the appropriate working tolerance, running the evaluation at a few different tolerances might be necessary to obtain a complete performance profile.

4.3.1 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 dilate 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 dilate the Derived Model to include all reference-segments within some tolerance area or volume around the Derived segment.

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:”

$$\frac{TP}{(TP + FN)} \quad (1)$$

It has a range from 0 to 1.0 (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.

$$\frac{TP}{(TP + FP)} \quad (2)$$

It has a range from 0 to 1.0 (a large value is good).

Branching Factor: The number of false-positive instances for every true-positive.

$$\frac{FP}{TP} \quad (3)$$

This metric can vary from 0 to infinity (a small value is good).

4.4 Progress in Road Modeling

A primary goal in the first year was to develop and demonstrate the technology necessary to reduce by an order of magnitude (factor of 10) over 1996 extraction practice, 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 McKenna Military Operations in Urban Terrain (MOUT) facility and the surrounding area at Ft. Benning, GA. This area contains approximately 20 km of roads over an area of 6.5 km² and, in the data set used for the work reported here, is covered by 44 frames of 1:5000 vertical panchromatic metric photography.⁵

The baseline performance benchmark of 279 minutes was established by an extraction task for the McKenna MOUT area. The task was performed by a professional cartographer using a digital stereo photogrammetric workstation (DSPW) running GDE SocetSet software [Goddard, 1996], as part of the 1996 High-Resolution Database Extraction study sponsored by the U.S. Army Topographic Engineering Center (USATEC). Thus, our goal the first-year demonstration was to extract the roads in 27.9 minutes.

In our formal demonstration held at SRI on 22 April 1998, we showed an integrated, end-to-end process (Figure 2) that produced a 3-D road network for the McKenna MOUT area (using the same images employed in the original benchmark extraction). The fully-automatic result ⁶ was 86% complete and had a correctness score of 90%. After interactive editing, these values were increased to 93% complete and 98% correct.

Approximately 25 minutes of human time was needed for “fixups” and quality control, resulting a model equivalent to the professionally produced one, with one-tenth the effort. ⁷

We have run the editing process three times for timing purposes. The human interaction time varied from 22 to 25 minutes, always under our goal of 27.9 minutes. The final scores presented above are not a perfect 100% because, as discussed earlier in this section, we cannot generally expect two human analysts to produce identical delineations. In this case there were locations in the imagery where portions of the road network are hidden under the tree canopy and the exact path was estimated differently by the humans preparing the reference and doing the final editing. There were also some minor errors in both the reference model and in our edited derived model.

⁵A new dataset that includes oblique as well as vertical coverage is in preparation.

⁶The automatic road-extraction process (the low-resolution process followed by the 3-D multi-image refinement process) took approximately ten minutes of computer time on an SGI O2 with a 175MHz MIPS R10000 processor. In more recent tests, the same processes took only five minutes on a Sun Microsystems Ultra30 with a 300MHz UltraSPARC processor.

⁷We also informally demonstrated the ability of the automated segment of the system to model roads appearing in 5m. CIB imagery taken from the Santiago dataset.

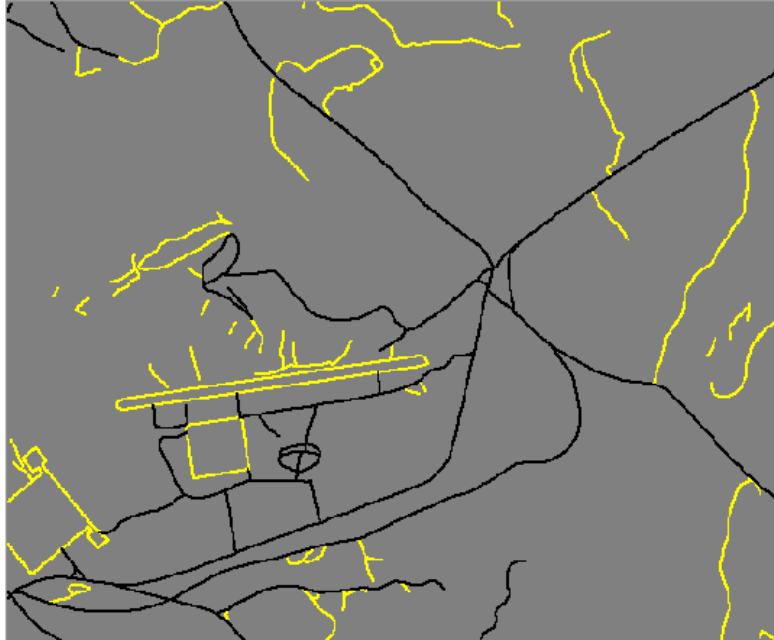


Figure 8: Ft. Benning reference road model. The black segments are the roads to be extracted; The yellow segments are the don't-evaluate segments.

Figure 8 shows the Ft. Benning reference road model, and Figure 9 shows the color-coded results of the evaluation process.

4.5 Virtual Laboratory Facilities for Road-modeling

The Virtual Laboratory allows users on the World Wide Web to access SRI's low-resolution linear delineation and evaluation algorithms. For linear delineation, users can fill out a form to specify the URL of a source image and algorithm parameters. When submitted, the Virtual Lab server runs the linear delineation algorithm and presents the results on a web page. Delineation results can then be used as input to a second form that allows evaluation of those results against a reference road set.

The linear delineation page of the virtual laboratory can be accessed at the URL <http://www.ai.sri.com/~apgd/vl/remote/ld-exp.html>, and the evaluation page can be found at the following URL <http://yuba.ai.sri.com/cgi-bin/lisp/evaluation.lisp>.

The evaluation page allows users to retrieve images and submit delineation results directly (without using SRI's delineation algorithm). This feature is intended to be used by those who wish to apply their own algorithms for delineation on SRI's images, and then evaluate their results using SRI's reference data.

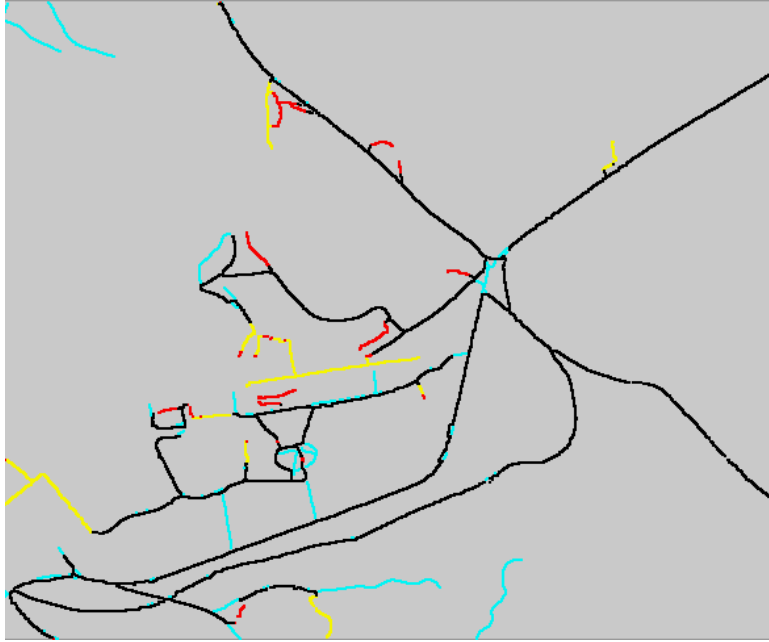


Figure 9: Color-coded road segments produced by the evaluation process. The black segments are the correctly located road segments; the blue segments are the false positives; the yellow segments are the extracted segments that match don't-evaluate segments; and the red segments are the false negatives.

5 BUILDING-EXTRACTION TECHNIQUES

Most current building-extraction systems [Gruen and Nevatia (Eds), 1998] are designed to find restricted classes of buildings, such as peaked-roof rectangular buildings or flat-roofed buildings with polygonal footprints. Although these systems are steadily improving, when presented with a general request, such as “find all buildings in this area,” they have the same problems as many other undirected extraction techniques – they hypothesize buildings where there are none and they miss some that are there. They produce extra hypotheses because the images contain configurations of features, such as lines, bright areas, and dark areas, that look like a building. For example, a road intersection with a crosswalk can look like a rectangular flat-roofed building. In addition, they miss buildings because they are partially occluded by trees or have confusing items on their roofs.

For this project, therefore, we have chosen a cue-point-based strategy for extracting buildings. In this strategy, a person (or program) specifies a list of building sites (cue points), and then the extraction procedure constructs the best building it can at each site. As a final step, we provide an interactive editing phase in which the user can make final adjustments, if necessary. Thus, our building-extraction process has three steps, which are similar to the three steps in road extraction:

Generate a list of cue points. This can be done interactively by a user or automatically by an analysis of elevation and other data.

Extract a building at each cue point. This step constructs a building model, working either bottom-up from 2-D and 3-D image patterns, using a technique developed at the University of South-

ern California (USC), or working top-down from parameterized 3-D models, using a technique developed at GDE.

Edit the extracted buildings. A person edits the results to produce a product that meets the application specifications.

During the first year of the APGD project, GDE enhanced a model-based technique that they had developed as part of an earlier effort and evaluated the building-extraction system developed at USC. The GDE technique starts with an initial building model and adjusts it by analyzing image regions and edges. The USC technique constructs 3-D building models from two or more images of a site by identifying patterns of linear features and shadows. The GDE technique requires a cue point and an initial model. The USC technique does not require a cue point or model, but it does significantly better given cue points.

GDE also developed, under its internal funds, a building “editor” for quickly deleting erroneous buildings, adjusting hypothesized buildings, and inserting new ones. GDE has used this system to interactively supply cue points and initial models to their extraction routines. (An executable-only version of this editor is available to the APGD program.)

Vexcel implemented a technique for automatically generating cue points and initial building models by analyzing dense digital-elevation models (DEM) and multispectral data. The basic idea is to use the dense DEM to identify 3-D lumps in the terrain that could be buildings, and then use the context provided by other available information to distinguish man-made lumps from vegetation. The system assumes that each man-made lump is a building.

These building-cueing and extraction techniques were evaluated by applying them to the Ft. Benning MOU site data and subwindows of the Ft. Hood data.

5.1 Cue Points and Building Models

A cue point is an point in object space that lies somewhere within the footprint of a building. Normally it is close to the centroid of the footprint.

We have defined three types of building models that can be specified at a cue point:

1. A polygonal footprint, often restricted to a rectangle.
2. A flat roofed building, represented by a polygonal footprint and an estimated height, often restricted to a rectangular footprint or L-shaped footprint.
3. A rectangular building with a gabled roof, represented by a rectangular footprint, a wall height, and a peak height.

The building-extraction routines use these initial models to focus their analysis on specific areas in the region of interest, and sometimes to initialize an optimization process that fills in and refines the model parameters.

5.2 Cue Point and Initial Model Generation

We developed two ways to generate a list of cue points and building models for a site. In the first method, a person interactively indicates points in images which are then automatically monoplotted against the terrain model to derive the object-space coordinates of the cue point. In the second, a program can automatically analyze data sources, such as high-resolution DEMs produced from optical imagery, interferometric synthetic aperture radar (IFSAR) data, and multispectral and hyperspectral data.⁸ GDE has implemented a technique of the first type. Vexcel has implemented a technique of the second type. In the GDE system, a person selects a building model from a menu of models, and then clicks on a few points in an image to specify the initial estimates for the building's parameters, such as location, orientation, and height. At present, the system supports three building types: a rectangular flat-roofed building, an L-shaped flat-roofed building, and a rectangular gable-roofed building. To create an initial model for a rectangular flat-roofed building, the user mouses three corners of the roof and one corner on the ground.

Vexcel's technique automatically generates cue points and initial building models by analyzing dense DEMs and multispectral data. The technique locates 3-D lumps in the terrain that could be buildings, and then uses all available information to distinguish man-made lumps from vegetation. The system assumes that each man-made lump is a building.

Given a terrain lump that has been labeled as a possible building, the Vexcel system currently estimates the position, orientation, and height of a rectangular flat-roofed building that surrounds the lump. It does this by fitting a rectangle around the footprint of the lump, and then setting the height of the building to the highest point in the lump.

Vexcel's technique was applied to the data at the Ft. Benning MOUT site. The data included:

- A set of four Sandia spotlight IFSAR images, taken by an aircraft flying along the four cardinal directions.
- Daedalus multispectral data.
- IFSARE imagery, including the raw SAR component images.

The four Sandia images were combined to form a single, dense DEM of the area. The height of each point in the combined image was computed by selecting the height with the highest radar correlation value in the four raw images.

All the data were used to classify points in the site into a few classes, including water, man-made object, dense vegetation, and field. The classification results highlighted the importance of being able to combine dense DEM information with multispectral data and radar data. When the classification system was run using just the IFSARE information, the buildings were identified as man-made – but so were many other things, especially bare tree trunks. When the classification was run using just the Daedalus data, the buildings were again labeled as man-made, but so were many other things, including rocky regions that looked like roofing material. Fusing the two

⁸A third possibility is to use a previous map of the area to supply building footprints or models of the buildings at an earlier time.

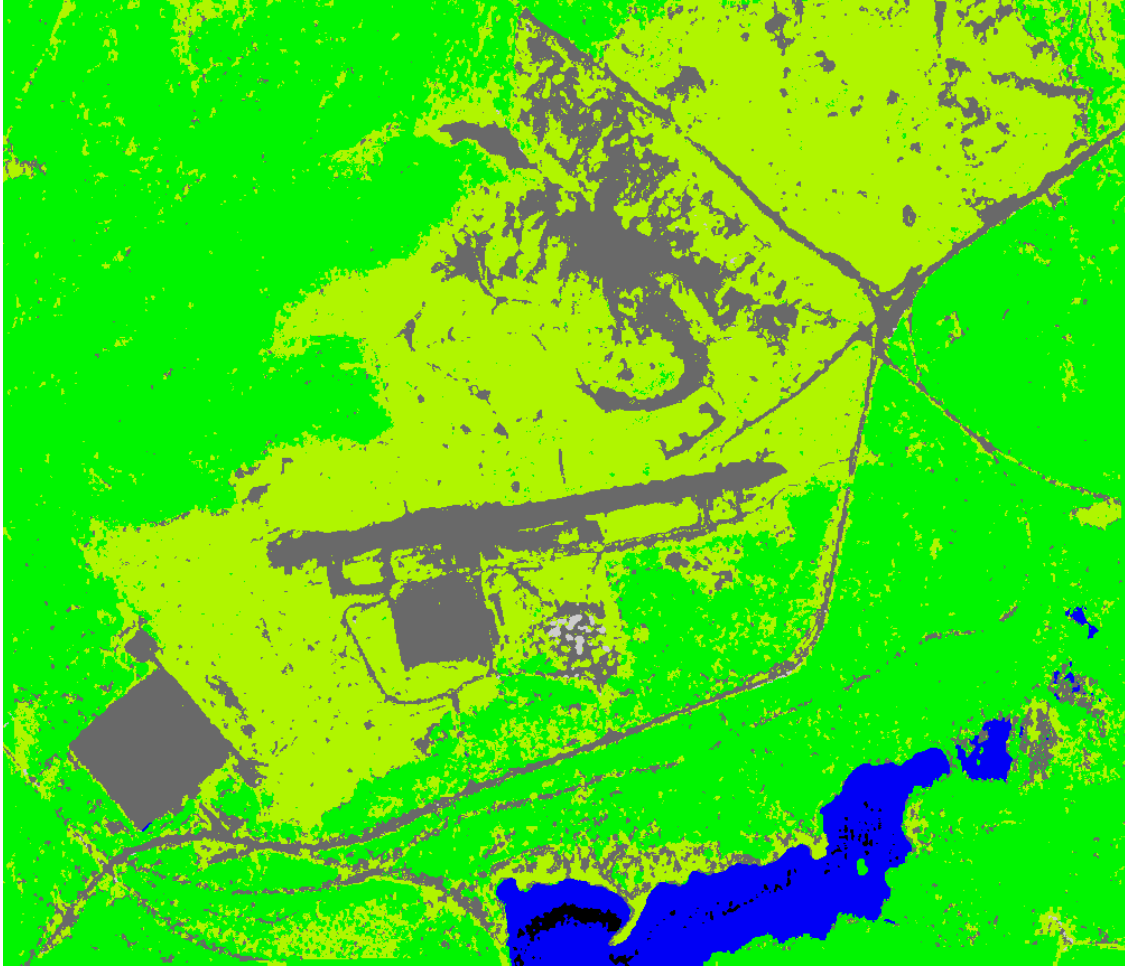


Figure 10: Fusion of the IFSARE and Daedalus classifications at the Ft. Benning McKenna MOUT site.

classifications (essentially “ANDing” them) eliminated virtually all false positives, enabling very reliable detection of buildings. Figure 10 shows the results of this fusion process.

At the time of the annual demonstration, Vexcel did not have an automatic way to use the classification results (shown in Figure 10), to distinguish buildings from dense vegetation (trees). Therefore, Vexcel automatically fit rectangles to all the lumps in the combined spotlight DEM, and then manually deleted the ones associated with trees. The results are shown in Figure 11. The evaluation of these results were

True Positives	13
False Positives	0
False Negatives	2
Don't Evaluate	0
Completeness	0.87
Correctness	1.00
Branching Factor	0.00

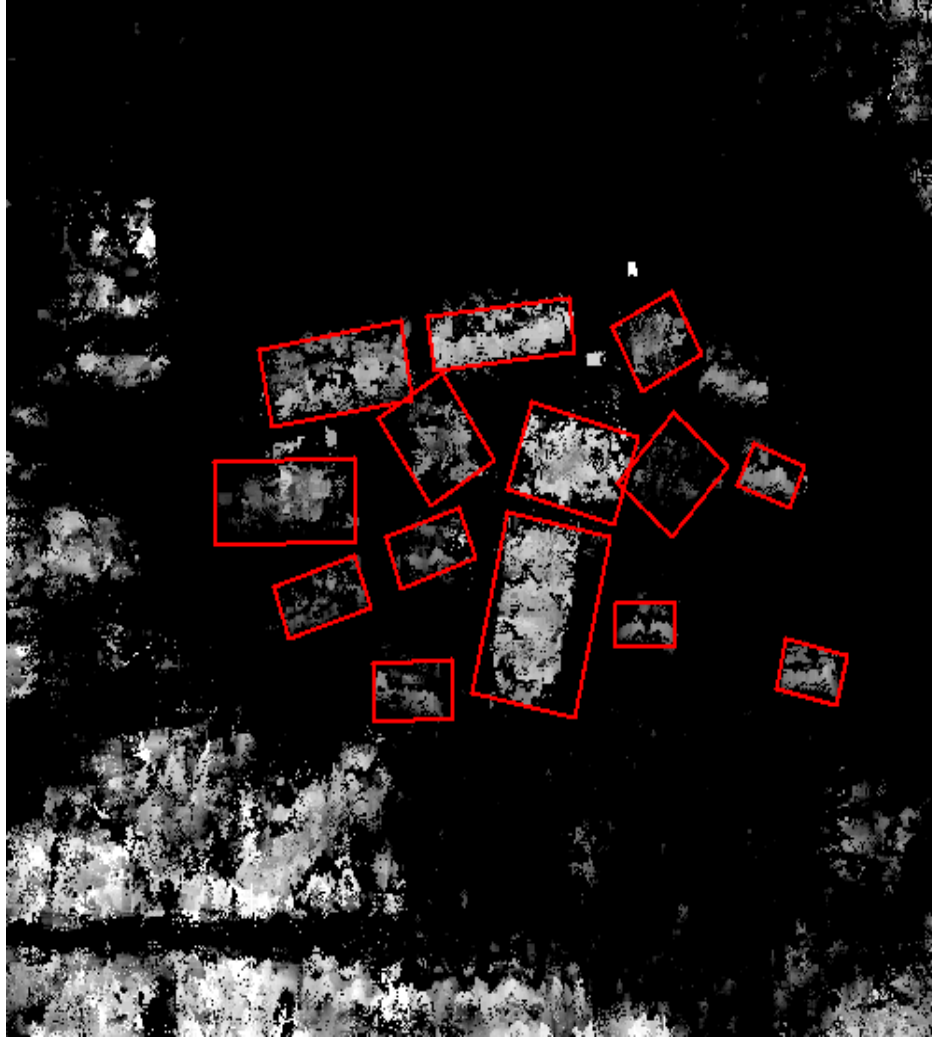


Figure 11: Rectangular footprints automatically fitted to lumps detected in the combined Sandia Spotlight IFSAR DEM and interactively filtered to include man-made objects.

The reference model contained 15 buildings, instead of the 19 used in some experiments, because the large building consisting of four gabled pieces was defined as one building, and the unusual shaped building was also defined to be one building, not two. By relaxing the requirements for a correctly extracted building, the system was able to extract 14 buildings, but in doing so, introduced a false positive. The evaluation results for that run were

True Positives	14
False Positives	1
False Negatives	1
Don't Evaluate	0
Completeness	0.93
Correctness	0.93
Branching Factor	0.07

Algorithm Block Diagram

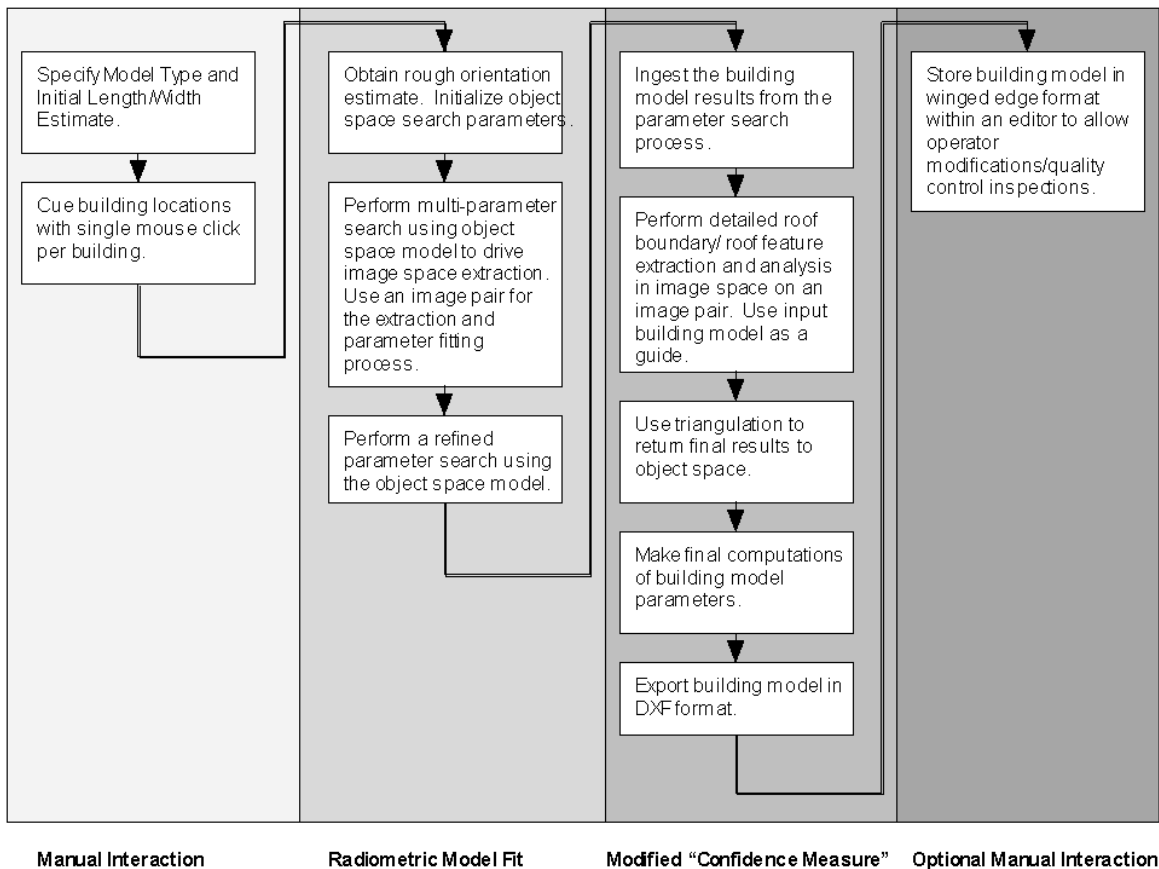


Figure 12: GDE building-extraction block diagram.

These results show that multispectral data and SAR data can be used to filter out lumps that are not buildings. Given just Daedalus data or just IFSARE data, the derived classifications include a large number of false positives (in the order of 10 to 40 times the number of building pixels). However, if both Daedalus and IFSARE data are available, the number of false positive drops dramatically, so that the number of false positives is approximately the same as the number of real positives.

5.3 Building Extraction, Given a Cue Point and Building Model

GDE's building-extraction technique operates in four phases, as shown in Figure 12. In the first phase, the user interactively selects building types and approximate parameters, then mouses points in the images to indicate where they occur. In the second phase, the system automatically adjusts the initial model parameters by analyzing intensity regions around the cue points. In the third phase, the system automatically refines the models produced by the second phase, adjusting the roof outlines so they line up better with edges extracted from the imagery. And finally, in the fourth phase, the user can interactively edit the results of the first three phases.

This procedure can extract three types of buildings: rectangular flat-roofed buildings, L-shaped



Figure 13: Buildings extracted by the GDE process applied to the McKenna MOUT site.

flat-roofed buildings, and rectangular gable-roofed buildings. It works best with imagery with a ground sample distance (GSD) of 0.5 to 1.0 meter. Its performance is reduced when applied to buildings that are moderately occluded, have poor contrast, or have a large amount of clutter on the roof.

At the end of the first year, the technique was tested on two sub-windows of the Ft. Hood data and the Ft. Benning MOUT site. The results on the MOUT site are shown in Figure 13. The evaluation of the results are shown in Figure 14.

The technique was also applied to the Ft. Benning MOUT site, starting with the output of Vexcel's semi-automatic cue point generation process. The evaluation of these results are shown below. They are lower than the interactively-initiated results because the semi-automatic cueing technique only produced cues for 14 buildings, grouping the four pieces of the large multi-gabled building into one model and grouping the two pieces of the odd shaped building into one model.

	Total Bldgs.	True Pos.	False Pos.	False Negs.
Ft. Hood 1	23	16	0	7
Ft. Hood 2	43	35	0	8
Ft. Benning	19	15	0	4
	Branching	Completeness	Correctness	Robust Completeness
Ft. Hood 1	0	70	100	70
Ft. Hood 2	0	81	100	81
Ft. Benning	0	79	100	79

Figure 14: Evaluation of GDE-extracted buildings.

True Positives	14
False Positives	1
False Negatives	5
Don't Evaluate	0
Completeness	0.74
Correctness	0.93
Branching Factor	0.07

These results show that current building-extraction techniques can correctly model relatively simple, isolated buildings. They need to be extended to handle occlusions and low-contrast situations, plus additional research is required to extract models of complex buildings.

6 CURRENT STATUS AND FUTURE PLANS

6.1 Status

6.1.1 Roads

Our effort to design a baseline linear delineation (LD) system as part of the BOS architecture, and to integrate it into the RCDE system for evaluation and testing, is complete. In addition to the semi-automatic Model-Based Optimization and correlation-tracking algorithms already resident within the RCDE, we have selected components from our low-resolution generic-LD-research-system (developed on Symbolics Lisp Machines) and re-implemented the code as needed for RCDE compatibility. Some errors were introduced in the conversion process, and these are now mostly eliminated. Some specific modifications and additions were needed to specialize the system to deal with roads, and these were inserted; in particular specialized filters to distinguish roads from other generic linear structures and linking techniques to generate a complete road network graph structure. We also modified the LD code to facilitate 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.

Our accomplishments detailed in this report represent important progress, but to achieve the ultimate goal of completely automated robust delineation of roads (streets, etc.) appearing in aerial images, we must solve additional problems.

- The most important problem we face in assembling a fully automated road delineation system, which has little or no need for final human inspection or 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 is obviously not present. This means that the system must be capable of a high degree of self-evaluation. We currently have context-based procedures that perform such evaluation but these procedures must be greatly expanded and made to return verifiable quantitative assessments (rather than just acting as go/no-go filters).
- 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 relate to streets. The notion of a simple, stand-alone algorithm to perform a complex recognition task is not viable. Our approach to structuring the APGD task within the framework of a context-based architecture potentially offers the feature-extraction algorithms easy and effective access to high-level contextual and semantic knowledge. We are just beginning to understand how to take advantage of this potential, and are now designing algorithms that can pose questions to the available knowledge sources to improve their performance. (For example, rather than requiring positive evidence, an algorithm can ask if there is any evidence preventing a road from bridging a gap. If not, it can fill in the gap.)

6.1.2 Buildings

Using a cue-point-based strategy together with a specialized interactive editor, as discussed in the preceding section, we were able to provide an end-to-end demonstration of building extraction that achieved our targeted goals for completeness, correctness, and maximum time required for human interaction (cueing and editing). We correctly modeled relatively simple isolated buildings in the Ft. Hood dataset and produced correct models for all the buildings in the McKenna MOUT site. We also demonstrated that fully automatic cueing is feasible if given the sensor data needed to construct a very high resolution DEM and a land-cover classification overlay of the area to be modeled. Additional work is required to extend these results to more difficult viewing conditions (e.g., occlusion, lack of shadows or excessive shadowing, low contrast) and more general and complex buildings.

6.2 The Year2 Feasibility Demonstration, January 1999

A major milestone in this APGD effort is to provide a formal “end-to-end” demonstration of progress in achieving our primary technical goals. In particular, extraction of all the roads and

buildings at the Ft. Benning McKenna Military Operations in Urban Terrain (MOUT) facility and the surrounding area, duplicating a professionally produced model with less than one-tenth the required human effort for buildings, and between 1-2 orders of magnitude reduction in human effort for roads.

In addition we plan to demonstrate:

1. Generality of the baseline extraction techniques

We will show results of rural and suburban road extraction on a collection of classified and unclassified datasets, and building extraction on unclassified imagery (including the newly acquired data from both Ft. Benning and Ft. Hood).

2. Advances in automated extraction using very high resolution, multispectral, and SAR/IFSAR data

For our baseline demonstrations, we plan to restrict our image and support data to standard NIMA products and National Imagery. However, we will also demonstrate opportunities for significantly enhanced automated extraction using currently available but non-productized imagery.

3. Demonstration of near-term technology transfer tools and techniques

We will demonstrate some self-contained extraction tools that could be transferred to NIMA for evaluation without major redesign or repackaging.

6.3 The Year2 High-Level APGD Work Plan

Our plans for Year2 are centered on advancing/completing our current development of the following major components of the APGD-IFD effort:

1. BOS/CBACS ARCHITECTURE

- (a) Implement the additional infrastructure and software components needed to allow employment of externally supplied *context* in all feature extraction operations. The use of context is critical to the development of highly competent feature-extraction systems. In Year2, we plan to employ context-based control of our road and building extraction techniques, and exploit context in the following ways:

- Provide 3-D tree canopy information to the rural road extraction procedures, so that they can predict occlusions and shadows.
- Use building and tree information to help detect suburban streets (and avoid misclassifying aligned houses or trees as roads/streets).
- Extend the competence of our building extraction techniques by employing contextual constraints provided by adjacent streets, trees, and nearby buildings.
- Enable the use of previously compiled maps and road construction practices to help detect and verify the presences of roads.

- (b) Complete the development of *editors* for road and building extraction. These editors are required to interface between the results obtainable from existing fully-automated extraction techniques and the requirements of potential users and established applications. The editors also make it possible and practical for the APGD-IFD team [SRI + GDE + VEXCEL] to create the reference models required for testing and evaluation.
- (c) Complete the software components needed to allow *inter-operability* between CME and SocketSet, and extend (as needed) our current *data interchange* format for APGD community-wide sharing of images, camera models, reference models, evaluations, and extraction results.

2. FEATURE EXTRACTION TECHNIQUES

- (a) Extend current techniques for rural *road* delineation to suburban streets. Improve performance, with respect to completeness and correctness, to 90+ percent for fully automated road delineation and a 50X reduction in the need for human intervention in rural environments, and 80+ percent for fully automated delineation and a 10X reduction in the need for human intervention in suburban environments.
- (b) Enhance the fully automated performance of the current (planar faced) building extraction techniques from current 50-70 percent level (completeness and correctness) to 80+ percent, and establish a 10X reduction in the need for human intervention.
- (c) Transfer new feature extraction techniques to the SCIF. Provide infrastructure for very large images and classified sensors.
- (d) Complete the evaluation of the use of *multisensor data* (primarily EO, IFSAR, and multispectral) in providing complementary information for road and building extraction algorithms. We have already established the utility of IFSAR and multispectral (e.g., Daedalus) in distinguishing between trees and buildings, and in providing other types of land-use-land-cover classification needed for context determination.
- (e) Extend our current work on modeling trees – as a critical contextual component in building extraction and street and road delineation – to allow more reliable detection and attribution:
 - Define a data model and data formats for trees
 - Develop techniques to translate existing data products into the tree data model
 - Develop algorithms to extract trees from stereo EO, IFSAR, and multispectral data
 - Extract and append tree data to the APGD-community data sets for Ft. Benning and Ft. Hood.

3. WWW VIRTUAL LAB

Update the virtual lab to include descriptions of the latest APGD results; add descriptions of the evaluation procedures and results.

4. DATA SETS, EVALUATION METHODOLOGY, and BENCHMARK EVALUATIONS

The APGD-IFD team has assembled comprehensive data collections and associated reference models of the roads and buildings at Ft. Benning and Ft. Hood suitable for conducting

comprehensive benchmark evaluations at these locations. We will support the APGD community in testing their algorithms and will use this data to quantify our own progress.

5. DEMONSTRATIONS and REPORTS

- (a) The APGD-IFD team recently completed its Year1 demonstration. Our plans for Year2 include major demonstrations at a classified meeting in November, the Image Understanding Workshop, a public APGD presentation at the end of January and a second APGD promotional meeting in February or March.
- (b) A video tape, that illustrates and documents the APGD feature extraction techniques developed in this first year has been included as part of this report; this video tape will be revised to include our further advances in year2.
- (c) A Year2 annual report will document our accomplishments in this Project.

6. TECH TRANSFER

We will work with NIMA to identify one or more suitable feature extraction techniques that could be transferred to their laboratories for evaluation and testing.

ACKNOWLEDGMENTS

We gratefully acknowledge the efforts of the following individuals who have contributed to our efforts, provided valuable insight, and helped keep the work focused on its intended goals: Doug Caldwell, Doug Climsonson, George Lukes, Michele Motsko, Lynn Quam, and Helen Wolf.

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 Heller, 1998] Martin A. Fischler and Aaron J. Heller. Automated Techniques for Road Network Modeling. 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 *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, 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/v1/datasets/Benning/db-report/>.
- [Gruen and Nevatia (Eds), 1998] A. Gruen and R. Nevatia (Eds). Special issue on automatic building extraction from aerial images. *Computer Vision and Image Understanding*, 72(2), November 1998.
- [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.

A SPECIFICATION FOR APGD DATASETS

For the purposes of APGD research, the primary components of a dataset are:

- A definition of a Local Vertical Coordinate System (LVCS) for the study area. This is a right-handed, Cartesian system, with the Z-axis normal to the reference ellipsoid, and the Y-axis pointing north. The origin should be chosen to be near the center of the study area or selected according to the criteria of the Global Coordinate System. Measurements relative to this origin are made in meters. This is used to facilitate data interchange between researchers.
- A collection of overlapping high-resolution panchromatic frame images of the study area (“the site”). For extraction of roads, vertical imagery is essential. For extraction of buildings, similar scale oblique coverage is needed as well. Since many IU algorithms that recover 3-D shape from panchromatic imagery have optimal performance with 4 or more, it is desirable to include this sort of coverage. These images should be 30 cm or lower GSD (approx. 1:5000 for vertical coverage). It is also desirable to have some imagery at lower resolution, 1m GSD, that covers the entire study area. The digitized images should put in Tagged Image File Format (TIFF), with a tile size between 128x128 and 1024x1024.
- When a metric camera is used, a copy of the camera calibration certificate and a diagram showing how to identify the fiducials should be included. For other frame cameras, any calibration information, how it was obtained (e.g., read off the lens barrel), and an estimate of the accuracy should be listed.
- Solutions for the interior and exterior orientations of these images arrived through a simultaneous bundle adjustment in a least-means-square framework, producing both the camera parameter and their covariances (“aerotriangulation”). If possible, the exterior orientation and covariances should be expressed relative to the LVCS. For frame camera photography, Extended USATEC format should be used.
- The image measurements and ground control used for the solutions, with photo-identifiable descriptions for the points. Global positioning system (GPS) readings for camera locations should be included if available.

These recommendations are meant to apply to distribution of frame photography. NIMA imagery products should be included in their standard formats (e.g., NITF 2.0 with RPC, DPPDB, CIB).
- A terrain elevation model covering the study area, either NIMA DTED, or elevation data derived from the imagery and put into DTED format. DTED2 should be used if available.

Additionally, a dataset can contain:

- Radar coverage, such as IFSARE. For interferometric coverage – all three bands, magnitude, phase, and correlation – should be included.
- Multi-spectral coverage, such as Daedalus or Hydice.

- Scanned maps, such as ADRG.
- Reference models for certain classes of features, such as roads, buildings (or building footprints), hydrology; area classifications, such as forest, built-up area. These models can be taken from existing data such as NIMA DTOP or USGS DLGs, or they can be extracted interactively by a cartographer using existing technology, such as the DPW.

On this project, two datasets have been used for most of the work: Ft. Hood and Ft. Benning/McKenna MOUT.

The panchromatic imagery component of the Ft. Hood dataset had been collected for the RADIUS Program in October 1993, scanned in early 1995 by Carnegie-Mellon University (CMU) and distributed in April 1995. The first set of usable camera parameters for this imagery was calculated by CMU and distributed by SRI in late June 1995. A second set of camera parameters that shifted the vertical datum for the ground control from the geoid to the WGS84 ellipsoid were distributed the following August. These data were distributed as part of the IUBA proposer's information package.

In June 1997, SRI augmented these data with IFSARE and NIMA DTED Level 0. In November 1997, CMU announced the availability of Hydice data for a swath covering part of the motor pool area. The WWW page, <http://www.ai.sri.com/~apgd/vl/datasets/Hood/ft-hood.html>, describes the dataset and how to obtain it.

The panchromatic imagery component of the Ft. Benning/McKenna MOUT dataset was collected as part of High-Resolution Database Generation Study carried out by TEC. In October 1995, the 44 frames of 1:5000 vertical photography were scanned and controlled by GDE, who also extracted models of the buildings and other features in the MOUT area as well as the roads and cart tracks visible in the entire collection. This is described in detail in a report available from <http://www.ai.sri.com/~apgd/vl/datasets/Benning/db-report>. In August 1997, GDE translated the camera parameters in to USATEC format and produced a terrain model and orthomosaic for the study area. These data were distributed by GDE to the APGD community in October 1997. Several errors were identified in the file format of the images and camera parameters. These problems were addressed by SRI and a distribution was made in January 1998. Since then, the dataset has been augmented with ERIM IFSARE, Sandia Spotlight IFSAR, and Daedalus multispectral coverage. This entire dataset is available at URL <http://www.ai.sri.com/~apgd/vl/datasets/Benning>.

We are continuing to eliminate deficiencies in both datasets and enhance them so that they meet the criteria outlined above.

B SENSOR GEOMETRIC MODELS

The dominant use of a sensor geometric model (often informally called “a camera model”) in the RCDE is to project 3-space coordinates to image coordinates required during wire-frame model rendering. To support smooth interaction with the object models, this projection must be fast (i.e., under $50\mu\text{s}$). For central projection cameras, the 3-D to 2-D projection can be accomplished with as few as 20 floating-point operations, taking less than $5\mu\text{s}$ on a Sun SPARCstation.

The RCDE can also accommodate images generated by dynamic sensors, in which the image formation process involves movement of parts of the sensor and/or the entire sensor platform itself, such as panoramic, “pushbroom” and “wiskbroom” sensors. Full mathematical models of sensors of this type typically require two orders of magnitude more computation than for a central projection camera. To address this problem, the RCDE has facilities to fit and use a piecewise polynomial approximation (commonly referred to as the *fast block-interpolation projection* or FBIP) to the full mathematical model, as well as being able to import and make use of rational polynomial function approximations (RPCs) generated by photogrammetric workstations such as GDE’s SOCET SET.

Since the RCDE must deal with a variety of image and sensor types, a very general interface framework was developed. The only explicit knowledge assumed about the sensor model is obtained from either its world-to-sensor projection function $P(X)$ or its sensor-to-world projection function $P^{-1}(U)$. The actual implementation of the projection functions is arbitrary, so long as specific mathematical properties are maintained. In particular, the projection functions P and P^{-1} are expected to be well behaved such that the local differential geometry expressed by the Jacobian matrix of the projection function characterizes a fairly large local neighborhood.

In general, there is no explicit knowledge of sensor position, sensor orientation, focal length, and so forth. We have found that most calculations that use these parameters in graphical user interfaces and IU algorithms can be replaced by counterparts derived from only the projection function and its Jacobian matrix. Furthermore, given two sensor models, there is no global concept of epipoles or epipolar geometry. However, in local neighborhoods, the epipolar relationship between two images can be computed from their Jacobian matrices.

The primary application program interface (API) functions are `project-to-view` and `project-to-world`. `Project-to-view` takes a 3-D point and returns the image plane coordinates of the projection of the point. Every time a 3-D object is drawn on top of an image, this function is called for each vertex of the object.⁹

`Project-to-world` is somewhat more complicated. It first computes the ray or curve¹⁰ in space that is defined by the locus of points that project to the given point in the image plane, and then computes the intersection of that ray or curve with a surface, such as a terrain model or a plane of constant elevation. If this intersection fails, the function returns `nil`; otherwise, it returns the point of intersection. This functionality can be exercised by using the mouse to click on an image being displayed by the RCDE. In response, the RCDE displays both the image coordinates and apparent

⁹A caching scheme, described later, eliminates recomputing the calls to `project-to-view` except when 3-D coordinates or camera models are modified.

¹⁰This occurs in the case of SAR imagery or when the 3-D coordinate system is non-Cartesian, as is the case with geographic coordinates.

3-D coordinates of the selected point.

One of the main advantages of this generic, sensor-model-independent API is that code can be developed and tested in an unclassified environment with unclassified imagery, and then used in classified imagery environments without significant modification. This ability was key to the transfer of IU technology from universities into the RADIUS Testbed System.

B.1 Central Perspective Camera

This is the standard eleven-parameter “pinhole” camera model. It can be directly instantiated from the internal and external parameters ($x, y, z, \omega, \phi, \kappa, \textit{principal-point-u}, \textit{principal-point-v}, \textit{focal-length}, \textit{skew}, \textit{aspect-ratio}$),¹¹ from an orthonormal 4×3 projection matrix, or as the result of a space resection of an image. The RCDE also includes facilities for decomposing arbitrary projection matrices (such as those arising from a direct linear transform resection) into standard parameters. In addition, there are a number of user interface functions that can be used to dynamically adjust projections to create synthetic views of the modeled scene. We have implemented procedures to instantiate a central perspective camera from parameters given in a USATEC format file or from a SocetSet support file.

B.2 Orthographic Projection

The RCDE handles orthographic projections as a special case of the central perspective projection, with the inverse of the focal-length, $1/f$, set to zero. These projections are commonly used to import orthorectified imagery into the RCDE or to produce 2-D “campus maps” from 3-D site models.

B.3 Fast Block Interpolation Projection

As mentioned earlier, the full math models for some sensor types are computationally so complex as to make them unsuitable for use in an interactive system. The solution used in the RCDE is to employ an existing (but too slow) implementation as a “black box” for generating tables for the new, faster (arbitrarily precise) approximation. In all of the known existing math models, it is relatively inexpensive to compute the ray in space corresponding to a given pixel. Projection from a point in 3-space to the image may be much more expensive.

The multiplane fast block projective model divides the image into rectangular blocks of some specified size, and defines a set of planes (usually three planes) of constant elevation in 3-space that span the range of elevations of the terrain. Thus 3-space is divided into volume cells defined by the paths of the camera rays in 3-space at each of the block corners in the image, and the elevation planes in 3-space. The approach makes no explicit assumption about the sensor and, furthermore, the coordinate system is not required to be Cartesian, allowing the direct projection from WGS-84 $\langle \textit{longitude}, \textit{latitude}, \textit{elevation} \rangle$ to image coordinates.

For each elevation plane of each block, eight parameters, (a, b, c, d, e, f, g, h) , define the projection

¹¹For most applications, *skew* is fixed at zero and *aspect-ratio* is fixed at 1.

of a 3-space point in that plane into image coordinates:

$$u = \frac{ax + by + c}{gx + hy + 1} \quad (4)$$

$$v = \frac{dx + ey + f}{gx + hy + 1} \quad (5)$$

An arbitrary point in 3-space point $\langle x, y, z \rangle$ is projected to image coordinates in the following steps:

- Determine which block contains the 3-space point. In general, this involves a search over many or all image blocks and the following nontrivial computation within each image block.
 - For each block corner, interpolate the corner position at the elevation of the given 3-space point. The interpolation formula is of the form

$$x = x_0 \frac{(z - z_1)(z - z_2)}{(z_0 - z_1)(z_0 - z_2)} + x_1 \frac{(z - z_0)(z - z_2)}{(z_1 - z_0)(z_1 - z_2)} + x_2 \frac{(z - z_0)(z - z_1)}{(z_2 - z_0)(z_2 - z_1)} \quad (6)$$

$$y = y_0 \frac{(z - z_1)(z - z_2)}{(z_0 - z_1)(z_0 - z_2)} + y_1 \frac{(z - z_0)(z - z_2)}{(z_1 - z_0)(z_1 - z_2)} + y_2 \frac{(z - z_0)(z - z_1)}{(z_2 - z_0)(z_2 - z_1)} \quad (7)$$

- Perform a point-in-polygon test to determine if the given 3-space point is within this block.
- For each elevation $\langle z_i \rangle$ in the block, compute the $\langle u_i, v_i \rangle$ projection of the 3-space point.
- Perform an n^{th} -order interpolation (n being the number of elevation planes) of the $\langle u_i, v_i \rangle$ image locations with z being the variable of interpolation. For three elevation planes we have

$$u = u_0 \frac{(z - z_1)(z - z_2)}{(z_0 - z_1)(z_0 - z_2)} + u_1 \frac{(z - z_0)(z - z_2)}{(z_1 - z_0)(z_1 - z_2)} +$$

$$u_2 \frac{(z - z_0)(z - z_1)}{(z_2 - z_0)(z_2 - z_1)} \quad (8)$$

$$v = v_0 \frac{(z - z_1)(z - z_2)}{(z_0 - z_1)(z_0 - z_2)} + v_1 \frac{(z - z_0)(z - z_2)}{(z_1 - z_0)(z_1 - z_2)} + v_2 \frac{(z - z_0)(z - z_1)}{(z_2 - z_0)(z_2 - z_1)} \quad (9)$$

The multiplane block projective interpolation scheme is expensive to compute, especially when the image block containing the 3-space point is not known. To improve performance, the following heuristics have been implemented:

- **Spatial and Temporal Coherence:** When projecting a new 3-space point, first try the image block of the last point projected. If that fails, re-project the point using the parameters from the current block. Repeat until the point projects into the correct block. This process usually converges in two iterations.
- **Vertex Cache:** When projecting a vertex of a CME object, save the projected image position in a cache (hash table) associated with that vertex. Subsequent projections of that vertex are avoided by using the cached image coordinates. Of course, the cache must be flushed if the 3-space coordinates of the vertex or the camera parameters are changed.

B.4 Rational Polynomial (RPC)

The sensor geometry may also be specified as third-order rational polynomials of the form

$$u = \frac{P_u(x, y, z)}{Q_u(x, y, z)} \quad (10)$$

$$v = \frac{P_v(x, y, z)}{Q_v(x, y, z)} \quad (11)$$

where u and v are normalized image coordinates and x, y, z are normalized ground coordinates.

The numerator polynomials contain 20 terms and the denominator polynomials contain 19 terms. There are an additional 10 normalization parameters. In an NITF 2.0 image these are given in a header field, with the ordering particular to a given imagery product (e.g., DPPDB, 200EAA). In addition, the RCDE can use an ASCII version specified in the interoperation specification in Appendix C. RCDE has facilities to robustly fit an RPC geometric models to other projection types as well as a regular grid of world to image correspondence points.

B.5 Composite

Any one of the preceding projections can have an arbitrary sequence of 3-D-to-3-D transformations prepended to it and an arbitrary sequence of 2-D-to-2-D transformations appended to it to form

a composite projection. Where possible, elements of the sequence are automatically collapsed to increase the computational efficiency of the overall projection. 3-D-to-3-D transformations are typically used to adjust nonparametric projections (e.g., FBIP) during image resections and bundle adjustments or to create dynamic views that retain the overall geometric qualities of the original projection. Nonlinear 2-D-to-2-D transformations are typically used to model the optical distortions in lenses. Linear 2-D-to-2-D transformations are used to define the transformation from 2d-world (original image) coordinates to displayed image coordinates, and from displayed image coordinates to screen coordinates.

C INTEROPERATION SPECIFICATION

C.1 Scope

The purpose of this appendix is to define a set of files and their content that allows SRI's CME to operate with images, sensor models, and elevation data generated by other vendor's systems. While it refers specifically to GDE SocetSet, in principle, any other vendor's system that can be made to produce these file should be able to interoperate with CME.

At this writing, it does not address the exchange of 2-D or 3-D feature data.

C.2 SocetSet to CME File Hierarchy

A separate directory hierarchy must be created to hold the following directories and files. The environment variable SS2CMER00T should be set to the top directory in this hierarchy.

Each SocetSet project is described by the following file hierarchy:

- `$SS2CMER00T/project-name`: Project Directory
 - `project-name` .PROJECT: Project Description File
 - `project-name` .DTED: Digital Terrain Elevation Data file for the project.
 - For each image pyramid,
 - * `image-name` .PYRAMID: Pyramid Descriptor File
 - * `image-name` .PROPS: Image Properties (optional)
 - * and one of:
 - `image-name` .TEC
 - `image-name` .RPC
 - `image-name` .CAMGRID

C.3 Project Description File

File Format Identifier: A single line of text containing the ASCII string:

```
SOCET2CME PROJECT DESCRIPTION VERSION 1
```

Project Name: A single line of text containing the ASCII name of the project.

Project Image Directory: A single line of text specifying a relative pathname for storing the image pyramids of the project. This pathname is relative to the directory specified by the environment variable SS2CMEIMAGES. **NOTE:** This directory does not need to be under the main hierarchy specified by SS2CMER00T, however is expected that it will be remain fixed for a given installation (i.e., not change on a per-project basis.)

Reference Ellipsoid: A single line of text containing an ASCII designator defining the reference ellipsoid for this project. This is usually the ASCII string WGS84.

Horizontal Datum: A single line of text containing an ASCII datum designator defining the the interpretation of geographic coordinates (lat, long, elev). This is usually the ASCII string WGS84.

Vertical Datum: A single line of text containing an ASCII datum designator defining the the interpretation of geographic coordinates (lat, long, elev). This is usually the ASCII string WGS84 or MSL.

Project Bounding Box: A single line of text containing the geographic coordinates of a bounding box containing the project specified as follows:

lat_min lat_max long_min long_max elev_min elev_max

For projects that extend across longitude = 0.0, is is permissible for long_max to be greater than 360 degrees. The specification is $long_max = long_min + long_extent$.

Project Origin: A single line of text containing the geographic coordinates of the project origin, which is usually chosen to be near the center of the project. The project origin is specified as a triple of double-precision reals

lat long elev

and is used to define the Local Vertical Cartesian Coordinate System (LVCS) used by CME. This is a right-handed, Cartesian coordinate system with the z -axis pointing up, y -axis north, and x -axis east. Measurements relative to the LVCS are assumed to be in meters. **NOTE:** Current practice with CME is to make the LVCS origin lie on the reference ellipsoid (i.e, $elev=0.0$).

Default Elevation Model This is a planar surface that is used to determine elevations that are outside the area covered by the DTED. It is specified by the four coefficients of the implicit equation for the plane in the LVCS.

A B C D

Where, $Ax + By + Cz - D = 0$, with $A^2 + B^2 + C^2 = 1$. Using this formulation, a constant elevation, say e , may be specified by

0 0 1 e

C.4 Example Project Description File

```
SOCET2CME PROJECT DESCRIPTION VERSION 1
ft-benning-2
ft-benning-2/images
WGS-84
WGS-84
WGS-84
```

```

32.3638 32.385 -84.8166 -84.7858 0.0 1000.0
32.370167 -84.805 0.0
0.0 0.0 1.0 235.6

```

C.5 Pyramid Descriptor File

Each PYRAMID descriptor file has the following form:

File Format Identifier: A single line of text containing the ASCII string:

```
SOCET2CME PYRAMID DESCRIPTION VERSION 1
```

Pyramid Type: A single line of text containing a unique ASCII name defining the geometric relationship between the pyramid levels.

In CME, GAUSS-2 is the code used to describe a 5×5 Gaussian convolution kernel, which has the following inter-image transformation:

$$x' = 2x + 1 \quad (12)$$

$$y' = 2y + 1 \quad (13)$$

where $\langle x, y \rangle$ are pixel coordinates of level i of the pyramid, and $\langle x', y' \rangle$ are pixel coordinates of level $i - 1$ (next higher resolution).

Another common pyramid transformation is BOX-2 which is a 2×2 uniform weighted convolution kernel, which has the following inter-image transformation:

$$x' = 2x + .5 \quad (14)$$

$$y' = 2y + .5 \quad (15)$$

If SocetSet image pyramids have different inter-image transformations, we need to define appropriate codes for them.

Top-image to Sensor Transform: The coefficients of a 2×3 matrix defining the transformation from pixel coordinates of the highest resolution image in this pyramid to line, sample coordinates of the sensor model.

The coefficients are supplied double float ASCII in a single line of text as follows:

```
A B C D E F
```

Where,

$$line = Ax + Bx + C \quad (16)$$

$$sample = Dx + Ey + F \quad (17)$$

$$(18)$$

and x, y are the image coordinates in the highest resolution image of the pyramid.

This transformation provides a concise specification of all possible image rotation, translation (due to cropping), and scale change in relation to the full frame image corresponding to the sensor model.

Image Pathnames: List of image pathnames, one per line of text, ordered from highest to lowest resolution. These pathnames are relative to the project-image-directory defined in the Project Description File.

NOTE: Currently supported image file formats are: IU-Testbed, GDE/Vitec, TIFF, and some types of NITF 2.0 images. All images *must* be tiled with tile sizes between 64×64 and 1024×1024 , with 128 to 256 providing optimal performance. The images in a given pyramid or project are not required to use the same file format.

C.6 Example Pyramid Descriptor File

```
SOCET2CME PYRAMID DESCRIPTION VERSION 1
GAUSS-2
1.0 0.0 0.0 0.0 1.0 0.0
4-7/image.g0
4-7/image.g1
4-7/image.g2
4-7/image.g3
4-7/image.g4
```

C.7 Image Properties File

[TBD]

C.8 Sensor Geometric Models¹²

C.9 “TEC Header” Files

The document *Digital Photogrammetric Compilation Package*, “*Example of an Image Header File*,” which is available from the URL <http://www.ai.sri.com/~apgd/docs/photogram.pdf>. A detailed specification for using the USATEC Header format for specifying central projection sensor models. **NOTE:** The TEC Header reader in CME requires that the header file use the same reference ellipsoid and datums as the project.

C.10 RPC Files

RPC files contain a human readable, double precision floating point form of the 200EA RPF coefficients for third-order rational polynomials describing the sensor geometry. The file format is as

¹²a.k.a. “Camera Models”

follows:

- **File Format Identifier:** A single line of text containing the ASCII string:

SOCET2CME GEO RPC FILE VERSION 1

- **Coordinate Offsets:** One line of text containing the following 5 double float numbers:

line-offset sample-offset x-offset y-offset z-offset

- **Coordinate Scales:** One line of text containing the following 5 double float numbers:

line-scale sample-scale x-scale y-scale z-scale

- **Polynomial Coefficients:** Four groups of 20 double float polynomial coefficients packed 5 per line as follows:

line-numerator (4 lines, 5 coefficients each)

line-denominator (4 lines, 5 coefficients each)

sample-numerator (4 lines, 5 coefficients each)

sample-denominator (4 lines, 5 coefficients each)

Blank lines are permissible following any of the described lines.

The offsets and scales correspond to those described in the 200EA document.

$\text{internal_var} = (\text{external_var} - \text{offset}) / \text{scale}$

C.11 Example RPC File:

```
SOCET2CME GEO RPC FILE VERSION 1
4364.0 4435.0 32.3691452475 -84.80596453115001 250.0
4364.0 4435.0 0.006510801600001059 0.007818218049997938 250.0

1.0349343866545456e-02 -5.8812408562497727e-03 1.3375065415797436e+00 -1.1794876653575967e+06 -6.7552380342135927e-08
-1.4614947318237129e-07 3.3618607211452820e-05 1.3128261830700684e-07 4.9788980334008455e-05 1.1506823666021591e-06
-1.2333487429829396e-07 3.8266823701855931e-10 -4.5830882687439760e-09 5.4629482022537378e-10 4.3440621743395409e-08
-9.0051006837926806e-09 -2.1062175383318785e-07 2.4912817082252832e-08 -6.2775048363128455e-06 1.1794876705974077e+06

1.0000000000000000e+00 -7.3660781267045972e-03 1.0874596595422019e-02 -3.3026789458505779e-01 5.6124766572626535e-07
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00 4.0183127409311905e-07 -3.3313732282011994e-09
0.0000000000000000e+00 -5.1271334201909408e-09 4.975894906088022e-09 0.0000000000000000e+00 3.6119145081008265e-10
9.5714662396260664e-10 0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00

1.7043296838313866e-02 -1.3413009934966014e+00 -5.7342744722848433e-03 -1.9479068607008150e+06 -3.2356739005134890e-03
-5.0625850837204972e-02 -2.0125301414610159e-04 -5.4841972789998516e-04 2.2566430099472045e-03 -1.4162676602811844e-05
3.5904098906575576e-04 -8.3430080963195179e-06 2.4423777829582427e-05 3.7254621569788454e-04 -1.7979781811839211e-05
-8.0529601880074209e-08 -2.3305055845052779e-06 -1.6701227285260495e-02 -7.0302603438258243e-05 1.9479068652876851e+06

1.0000000000000000e+00 -9.0482576381464112e-03 1.3365339711521380e-02 -2.9667015648456652e-01 -5.4470935931232098e-05
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00 1.9015297209964444e-05 4.0748557461322360e-05
0.0000000000000000e+00 -5.3427613981254204e-08 2.1034549738275259e-07 0.0000000000000000e+00 -3.0512360933003033e-07
1.4590323912524047e-07 0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00
```

C.12 CAMGRID Files

CAMGRID files contain ground to image correspondences computed from a rigorous implementation of the sensor model on a regular grid in image space. It is to be used for sensor models that cannot be adequately characterized by a nine-parameter central projection or RPC.

File Format Identifier: A single line of text containing the ASCII string:

SOCET2CME GEO CAMERA GRID VERSION 1

Correspondence Data: Each remaining line in the CAMGRID file consists of 5 double floats as follows:

latitude longitude elevation line sample

Image and elevation coordinates must be gridded according to these criteria:

- There should be at least 8 values of line number spaced evenly between the minimum and maximum values line number in the image.
- There should be at least 8 values of sample number spaced evenly between the minimum and maximum values sample number in the image.
- The line number interval should not exceed 1024 pixels.
- The sample number interval should not exceed 1024 pixels.
- There should be at least 3 values of elevation spaced evenly between the minimum and maximum values of terrain elevation in the project area.

C.13 Example CAMGRID File:

```
SOCET2CME GEO CAMERA GRID VERSION 1
32.3627602818 -84.7982058824 0.000 0.000 0.000
32.3643145863 -84.8000908450 250.000 0.000 0.000
32.3658687399 -84.8019757244 500.000 0.000 0.000
32.3643477202 -84.7981985422 0.000 1091.000 0.000
32.3655077629 -84.8000853525 250.000 1091.000 0.000
32.3666676859 -84.8019720632 500.000 1091.000 0.000
32.3659415690 -84.7981911721 0.000 2182.000 0.000
32.3667057579 -84.8000798377 250.000 2182.000 0.000
32.3674698583 -84.8019683871 500.000 2182.000 0.000
32.3675418670 -84.7981837719 0.000 3273.000 0.000
32.3679086004 -84.8000743004 250.000 3273.000 0.000
32.3682752766 -84.8019646961 500.000 3273.000 0.000
32.3691486535 -84.7981763414 0.000 4364.000 0.000
32.3691163199 -84.8000687405 250.000 4364.000 0.000
32.3690839605 -84.8019609901 500.000 4364.000 0.000
32.3707619679 -84.7981688805 0.000 5455.000 0.000
32.3703289462 -84.8000631579 250.000 5455.000 0.000
32.3698959301 -84.8019572689 500.000 5455.000 0.000
32.3723818503 -84.7981613889 0.000 6546.000 0.000
32.3715465092 -84.8000575524 250.000 6546.000 0.000
32.3707112054 -84.8019535325 500.000 6546.000 0.000
32.3740083406 -84.7981538665 0.000 7637.000 0.000
32.3727690392 -84.8000519239 250.000 7637.000 0.000
32.3715298065 -84.8019497809 500.000 7637.000 0.000
32.3756414795 -84.7981463131 0.000 8728.000 0.000
32.3739965665 -84.8000462723 250.000 8728.000 0.000
32.3723517539 -84.8019460138 500.000 8728.000 0.000
32.3627448024 -84.8001103222 0.000 0.000 1108.750
32.3643029327 -84.8015223161 250.000 0.000 1108.750
32.3658609242 -84.8029342479 500.000 0.000 1108.750
32.3643343530 -84.8001068483 0.000 1091.000 1108.750
32.3654976969 -84.8015197236 250.000 1091.000 1108.750
32.3666609331 -84.8029325244 500.000 1091.000 1108.750
32.3659303312 -84.8001033603 0.000 2182.000 1108.750
32.3666972923 -84.8015171205 250.000 2182.000 1108.750
32.3674641771 -84.8029307939 500.000 2182.000 1108.750
32.3675327760 -84.8000998581 0.000 3273.000 1108.750
32.3679017483 -84.8015145068 250.000 3273.000 1108.750
32.3682706758 -84.8029290563 500.000 3273.000 1108.750
32.3691417267 -84.8000963414 0.000 4364.000 1108.750
32.3691110946 -84.8015118825 250.000 4364.000 1108.750
```

32.3690804490	-84.8029273116	500.000	4364.000	1108.750
32.3707572231	-84.8000928104	0.000	5455.000	1108.750
32.3703253609	-84.8015092473	250.000	5455.000	1108.750
32.3698935167	-84.8029255599	500.000	5455.000	1108.750
32.3723793053	-84.8000892648	0.000	6546.000	1108.750
32.3715445773	-84.8015066014	250.000	6546.000	1108.750
32.3707098990	-84.8029238009	500.000	6546.000	1108.750
32.3740080135	-84.8000857046	0.000	7637.000	1108.750
32.3727687742	-84.8015039446	250.000	7637.000	1108.750
32.3715296163	-84.8029220347	500.000	7637.000	1108.750
32.3756433884	-84.8000821298	0.000	8728.000	1108.750
32.3739979822	-84.8015012768	250.000	8728.000	1108.750
32.3723526890	-84.8029202613	500.000	8728.000	1108.750
32.3627292524	-84.8020199006	0.000	0.000	2217.500
32.3642912314	-84.8029576499	250.000	0.000	2217.500
...				
32.3756543172	-84.8118093016	0.000	8728.000	7761.250
32.3740061994	-84.8103156902	250.000	8728.000	7761.250
32.3723581931	-84.8088222499	500.000	8728.000	7761.250
32.3626344459	-84.8135864546	0.000	0.000	8870.000
32.3642200056	-84.8116516480	250.000	0.000	8870.000
32.3658054104	-84.8097169252	500.000	0.000	8870.000
32.3642389372	-84.8136106385	0.000	1091.000	8870.000
32.3654259999	-84.8116698000	250.000	1091.000	8870.000
32.3666129393	-84.8097290627	500.000	1091.000	8870.000
32.3658499784	-84.8136349220	0.000	2182.000	8870.000
32.3666369176	-84.8116880267	250.000	2182.000	8870.000
32.3674237648	-84.8097412500	500.000	2182.000	8870.000
32.3674676098	-84.8136593057	0.000	3273.000	8870.000
32.3678527888	-84.8117063284	250.000	3273.000	8870.000
32.3682379072	-84.8097534873	500.000	3273.000	8870.000
32.3690918718	-84.8136837902	0.000	4364.000	8870.000
32.3690736439	-84.8117247056	250.000	4364.000	8870.000
32.3690553869	-84.8097657751	500.000	4364.000	8870.000
32.3707228054	-84.8137083762	0.000	5455.000	8870.000
32.3702995136	-84.8117431588	250.000	5455.000	8870.000
32.3698762244	-84.8097781135	500.000	5455.000	8870.000
32.3723604518	-84.8137330643	0.000	6546.000	8870.000
32.3715304290	-84.8117616884	250.000	6546.000	8870.000
32.3707004405	-84.8097905029	500.000	6546.000	8870.000
32.3740048524	-84.8137578551	0.000	7637.000	8870.000
32.3727664212	-84.8117802950	250.000	7637.000	8870.000
32.3715280561	-84.8098029437	500.000	7637.000	8870.000
32.3756560491	-84.8137827492	0.000	8728.000	8870.000
32.3740075216	-84.8117989790	250.000	8728.000	8870.000
32.3723590922	-84.8098154361	500.000	8728.000	8870.000

D PROPOSED APGD EVALUATION PROCEDURES

This appendix ¹³ documents our attempt to establish a consensus on evaluation metrics and procedures among the members of the APGD research community. At present, it solely represents the views of its SRI authors. We expect there will be other positions preferred by other members of the research community. Nevertheless, it serves as a starting point for discussion and initial experiments.

In the evaluation process, components of a derived model will be compared with those of a provided reference model and will be evaluated with respect to a set of basic metrics (plus additional metrics that may be appropriate for specific features or applications). Multiple tests may be required depending on the purpose of the evaluation; for example, since most algorithms have the ability to redistribute their error budget, it may be informative to run the algorithm at least twice to highlight performance in a stand-alone mode where completeness is essential versus the ability of the algorithm to serve as a component in a composite system where predictability and robustness are of primary concern.

We employ the following definitions and metrics, which we call the General Model Evaluation Metrics (GMEM):

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.

Table 1: Definitions of quantities tabulated for General Model Evaluation Metrics (GMEM).

TRUE POSITIVE	TP	Reference and Derived models agree.
FALSE POSITIVE	FP	Found in the Derived model only.
FALSE NEGATIVE	FN	Found in the Reference model only.
DON'T EVALUATE	DE	Included in the Reference model in the DON'T EVALUATE class, but not considered to be either correct (TP) or incorrect (FN) when included or not included in the Derived model.

As discussed later, we have introduced a DON'T EVALUATE category to deal with ambiguous situations where cartographers might differ on whether or not an item should be included in the reference data.

From the 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

¹³Appendix E contains explicit specifications for the data models and ASCII file formats for roads and buildings.

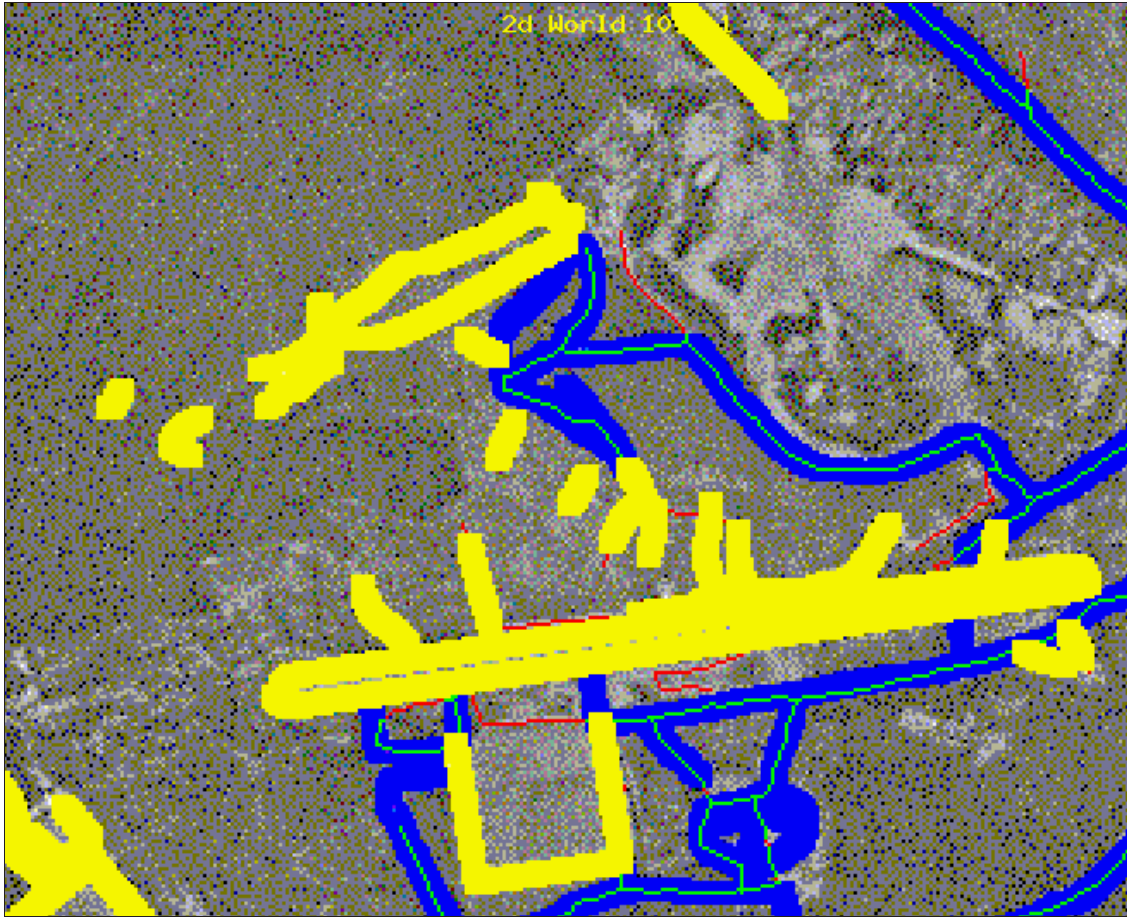


Figure 15: An example of the scoring of a road network extraction result. The TRUE POSITIVE reference model is shown in blue. The DON'T EVALUATE reference model is shown in yellow. Roads scored as TRUE POSITIVE are shown in green. Roads scored as FALSE POSITIVE are shown in red.

“detection percentage:”

$$100 \times \frac{TP}{(TP + FN)}. \quad (19)$$

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.

$$100 \times \frac{TP}{(TP + FP)}. \quad (20)$$

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

Branching Factor: The number of FALSE POSITIVE instances for every TRUE POSITIVE.

$$\frac{FP}{TP}. \quad (21)$$

This metric can vary from 0 to infinity (a small value is good).

D.1 Report Format for Describing an Extraction Algorithm and the Results of Applying It to a Dataset

This is information a researcher is encouraged to provide to describe an algorithm and its results.

1. General Philosophy of Approach and Specific Techniques Employed

This section should provide a general summary of the approach equivalent to the abstract for a scientific journal publication. It should include a block-diagram description of the major system components and key algorithms. It should also include a list of references (preferably annotated) to relevant published papers.

2. Qualitative Description of the Competence of the Algorithm

In addition to the specific information and formats required for the external automated evaluation process described below, qualitative information of the type specified in the following examples should also be provided where possible:

Algorithm Function: Extracts models of planar-faced buildings.

Algorithm Restrictions: Poor performance when trees are adjacent to buildings.

Algorithm Data Requirements: 15 cm or better GSD EO required; availability of SAR improves performance.

Algorithm Expected Performance Under Favorable, But Realistic, Conditions: 80% completeness and 95% correctness.

Algorithm Self-Evaluation: A relative self-evaluation function is employed internally to adjust parameters; dubious outputs are flagged.

Algorithm Automation Classification: Level IV (see below); each building to be modeled requires a Cue-point to be provided by some external source or a human operator as an initialization step.

Algorithm Maturity: Has been tested on hundreds of images and is now considered stable.

Geometric Info: Z values are obtained by dropping the roof outlines to a DEM.

Topological Info: Winged edge topology with nodes, arcs, and faces is maintained internally.

Attribute Info: Roof material type is determined for internal use.

Image Input Formats Readable: TIFF

Model Output Formats Available: DXF, VPF, and ARC/INFO.

3. Degree of Automation and Requirements for Operator Interaction

The degree of automation employed by the researchers running the test data should be identified (using the classification scheme presented below). A brief description should be provided of the various interactions required, including initial parameter set-up and any interactions with the algorithm while it is running.

We define the following major levels of automation for classifying APGD algorithms and systems:

Level-1: Able to extract the target object/feature in an image without human access to the image being processed or to similar images taken in the same area, or prior knowledge of the specific site under consideration. The algorithm can take advantage of acquisition and sensor parameters (e.g., weather, camera parameters, ground resolution), and general knowledge of the area being mapped/monitored (e.g., terrain type, terrain elevation, season, urban/rural, types of roads or buildings to be expected, etc.). Nominally, in a production system, this information will automatically be extracted from ancillary sources and will not require user interaction.

Level-2: Same as Level-1, but a human operator can tune the algorithm using images of areas similar to, but displaced from, the one being processed.

Level-3: Same as Levels 1 and 2, but the prior images can cover the area to be mapped/monitored.

Level-4: In addition to the above, a human operator can spend a brief amount of time initializing his algorithm using the actual image being processed. (This will have to be better quantified, but nominally, less than 10% of the time it would take for him to complete the delineation using the best available interactive tools).

Level-5: In addition to the above, a human operator can spend some time editing the derived model.

Level-6: In addition to all the above, a human operator can continuously interact with his algorithm in the delineation process.

4. **Timing and System Resources**

One of our primary evaluation metrics requires quantifying the amount of time the human operator spends on each task required to construct the final model, both actual interaction time and total “wall clock” time.

While less important, we also want to document the system resources required for the automated feature extraction, including computer clock time and the type of hardware employed.

5. **Road Extraction or Building Extraction Results**

Provide the extracted features in the specified format.

6. **Graphical or Pictoric Display of the Derived Models**

Display the derived models on an orthophoto or in 3-D as a stereo model.

D.2 **The Road Evaluation Process**

An extracted road model is evaluated at three levels: segment centerline location, segment attributes, and network topology. These evaluations are performed in the following ways, using the GMEM (General Model Evaluation Metrics) metrics described earlier.

D.2.1 **Segment Geometry**

The procedure for evaluating centerlines at the sample-point level involves projecting the centerline points into a provided orthoimage and comparing them pixel for pixel with a 3-D reference model

which has been projected in the same manner. We expand the width of the projected reference model roads to provide a tolerance for the extracted segments. We add an additional road-width to each side of the road (i.e., tripling the reference width) based on the assumption that the image-analyst might have positioned the reference centerline anywhere within the visible width of the road as it appears in the imagery, that there may be some quantization error due to the pixel-based representation, and there might also be some projection error if the image-analyst extracted the roads in 3-D using a stereoscope. The extracted roads are then intersected with the reference ribbon and extracted road segments falling completely outside the ribbon are scored as FALSE POSITIVE. Road segments that are both within and outside the boundary are clipped at the boundary; the lengths of the segments falling outside the ribbon boundary are added to the FALSE POSITIVE total and road segments falling within the ribbon are snapped to the reference model path and scored as TRUE POSITIVE. The dimensions of this metric are in terms of Km of road.

D.2.2 Segment Attributes

- Each of the segment attributes (that is, the summary segment attributes, not the individual centerline point attributes) are compared to the attributes in the reference model. For each attribute we obtain a percent-correctness score [number of segments with a correct value for the given attribute divided by the total number of segments].
- We also propose a metric that we call the average-connectivity-interruptions per Km. We compute this as a summation of [One less than the total number of correctly positioned road sub-segments found by the algorithm for each road segment in the reference], divided by a summation of [the lengths of all the reference road segments]; the dimensions of this metric are average-connectivity-interruptions per KM.

D.2.3 Network Topology

The GMEM metrics will be applied to the collection of network vertices/intersections (we will ignore intersections associated with driveways, spurs, etc.). A vertex is correctly modeled if its location and connectivity with respect to the incident roads agree with the reference model.

D.3 The Building Evaluation Process

A building instance is evaluated separately with respect to its three components (cue-point, footprint, roof-model). Each component is evaluated in terms of the “vertices” that define the component (see Appendix E.5). If any vertex of a component is found to be in error, either in location or connectivity, the component is considered to be incorrect. A geometric tolerance is applied to the location estimates. For buildings, which are represented by the 3-D location of the vertical-wall and planar roof surface intersections, we will accept a 6 pixel x-y deviation in the highest resolution (stereo-pair) imagery provided. The rationale for choosing these tolerances is based on the observation that manual and semi-automatic feature extraction techniques (e.g. RCDE and MBO systems) can typically localize features to within 1 to 2 pixels. In the Ft. Benning panchromatic imagery, where the relative errors in the camera parameters are negligible, this corresponds (in

object space) to a 15 to 30 cm horizontal error and a 1 to 2 meter vertical error. These figures are corroborated, by comparisons of the extracted building in the reference models to the actual construction plans of the buildings. We place the acceptance criteria for automatic extraction at three times this limit or 6 pixels in the image plane and corresponding distances in object space (i.e., 1 meter horizontal and 6 meters vertical), which roughly corresponds to a just noticeable difference to the eye when the model is superimposed on the image.

We apply all the GMEM metrics to the three components.

D.4 APGD Evaluation Philosophy and Rationale

Evaluation can be done for different purposes and the methods of evaluation may differ significantly depending on the purpose. The three types of evaluation we are primarily concerned with in the APGD effort are:

1. To describe progress in developing algorithms and systems to sponsors and peers, and to validate claims made in a specification of the algorithm's performance (required for use in integrating the algorithm into a composite system with more general capabilities than that provided by the individual algorithms):

Evaluation measures to describe progress may be very specialized and unique to each algorithm and problem domain. For example, special experiments and associated metrics might be needed to show the progress in road extraction contributed by learning algorithms. More generally, algorithms may be designed to model a limited class of features (or "sub-features"), e.g., paved roads. To quantify progress in paved road modeling, and to validate claims for the algorithm's expected performance, extracted features should be compared against paved road reference data.

In a benchmark exercise, a participant is encouraged to include additional metrics (with an explanation) that are relevant to his approach, but were not included in the basic evaluation plan.

2. To measure performance relative to a given application, or a set of user requirements:

Since current automatic algorithms are unlikely to extract all the features for a specific task, such as generating a 1:50,000-scale topographic line map, a systems-level application-oriented method of evaluating an algorithm is to measure the amount of human effort, (measured in analyst interaction time), needed to satisfy the user's product specification. For this type of evaluation, we plan to measure the time required to initialize the system and to edit the model that was produced by the system (we define editing to include any human interaction with the system after it has been initialized). Since (by definition) the final product meets the user's specification, the metric of primary interest is interaction time. (Never the less, in general, other standard metrics will still be collected and all processes timed.)

A less direct way of producing a similar result would be to (1) categorize the types of editing steps to be performed, (2) estimate a time required to perform an editing step of each type, and (3) total the estimated times for the given (specific) task. This approximation is less reliable than measuring editing time, but it may provide useful relative results.

3. To evaluate the relative utility of algorithms in performing a specific modeling task and to identify/diagnose strengths and weaknesses relative to this task:

Evaluation measures to compare the relative utility of algorithms nominally competent to perform a given task (e.g., measure road width) would be given test cases with known difficulties (e.g., a road that changes from two lanes to three lanes for passing) and their performance on such very specialized cases could be used to select a "complementary" set of algorithms that are sufficient to cover a known set of problems that no single algorithm can handle by itself. This approach could use properties of the modeled features in the reference dataset to form specific testsets. For example, a user could easily set up evaluations to (1) measure road width (for all types of roads), (2) measure the width of dirt roads, and (3) measure the width of roads partially occluded by nearby trees.

D.4.1 Discussion of Critical Issues and Assumptions

1. What is the "correct" answer; that is, what should appear in the reference model, and with what precision must it appear in the derived model.

The earth's surface is continually changing; an image collection acquired a few months ago, and used for benchmarking experiments, may no longer accurately describe what is currently on the ground. The algorithm that runs on a given set of images can only be expected to accurately describe the content of the available imagery (tempered by ancillary information, physical constraints, etc.). In most of our discussion we employ the term "Reference Model" to denote the nominally correct description that our extraction algorithms are expected to recover. Generally, the reference model is obtained by a human analyst modeling scene content from the imagery to be used in the benchmark tests.

One might expect that a comprehensive definition of the various features of interest (e.g., buildings and roads) is a necessary first step in evaluating the performance of feature extraction algorithms and systems. We assert that from a practical standpoint, it is impossible to provide a comprehensive operational/computational definition of something with instances as geometrically diverse and complex as a "road" or a "building" – that can be used as for evaluating correct vs. incorrect algorithm performance based strictly on image content. In fact, if such a definition was provided, it could be converted into a computer program that correctly performed the required task.

We note that dictionary definitions of buildings and roads are primarily concerned with their use, rather than their geometric structure or appearance; and even if it were possible to provide the desired definitions, there will always be a significant number of instances that are ambiguous with respect to their correct classification. For example, how do we geometrically define what a building is in a "bombed-out" city – and especially one that is inhabited and is being rebuilt. Can a moving/movable object be a building (what about a houseboat or a trailer; or even a trailer moving on a freeway)?? 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 road very close-by (say

over an open divider strip), should we insert an intersection at such a location even though it is "illegal" to cross over?? While some of the examples we have just listed are extreme cases, and most of the modeling problems we will encounter are obvious with respect to the correct interpretation, we need some way to avoid having to consider these extreme cases or allowing them to distort the true performance of an algorithm being evaluated – the use of a DON'T EVALUATE category (discussed below) offers a simple way of accomplishing this goal.

An algorithm is an implied computational definition of the feature it is intended to model. The algorithm designer usually bases his design on (1) requiring the 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., the roads in San-Francisco can be assumed to all be paved rather than dirt roads. The potential customer/consumer of the model probably has in mind a use-based (dictionary style) definition of the features in the model – e.g., a road is physical structure that facilitates the movement of vehicles, and indeed, is used for that purpose. The human image-analyst tries to use both types of definitions in his modeling, 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 are willing to defer to the customer/consumer's definition, we still have the problem that the image doesn't usually provide a way to establish if the definition is (or is not) satisfied.

Thus, since it will generally be impossible to provide a universal or comprehensive definition of the features of interest, it will also generally be impossible to build a general purpose feature (road or building) modeling system that is effective for all environments and applications. Further, there is no authoritative way to determine if the reference model produced by a human image-analyst is both complete and correct, and therefore provides a fair and neutral basis for evaluation – although one would expect that most of the instances encountered in realistic images would have obvious interpretations.

We deal with the above problems using two mechanisms. First, from a practical standpoint, in our modeling system, we will provide a means for a human operator to efficiently edit the automatically generated model to bridge the gap between what the algorithm designer used as an implied definition of the features, and what a specific customer desires for his application. Second, to deal with unavoidable instances of ambiguous features (either because of problems with incomplete definitions, or because the necessary information needed to make a proper decision is not visible in the available imagery), we must be able to label such instances as DON'T EVALUATE and exclude them from the evaluation process.

Finally, there is the issue of acceptable tolerance on the precision of modeled structures. The tolerance acceptable to a particular user has no effect on what is actually possible given a particular set of images (although it could well alter the choice of algorithms selected by the feature extraction system). In the evaluation process, the images and associated camera models are typically "given's," their errors should not be commingled with those of the algorithms being tested. Similarly, the errors made by the image-analyst in preparing the reference model are difficult to quantify, but should not be attributed to the algorithm per-

formance. As noted in item 4 (below), what we really want to know (and validate) is the precision specified by the algorithm designer for his product.

2. The cost of automation is equal to the cost of editing (correcting errors).

We assume that the computer cost and time required to run a typical image extraction algorithm on an image will continue to decrease, and will be insignificant within the five-year time frame of the APGD program. Thus, in a practical setting, the cost of automation largely amounts to the time spent fixing the errors and short-comings of the automated process. If the fix-up time for automated site modeling is more than the time it would take to manually extract the visible features, then there's no point to the "automatic" process. The time spent correcting an error in the output of an automated feature-extraction algorithm can be used as a weight on the importance of that error. With respect to geometric extent, a few small isolated errors that each require a specific action to correct can be more time-consuming to edit than a single very large incorrect coherent construct that can be fixed by a single action.

3. The utility of an algorithm is system, task, and implementation specific; algorithm evaluation can be generalized by categorizing and parameterizing the errors. Narrow object categories (e.g, distinguishing between simple roads, divided highways, and streets, rather than just the single category of roads) provides more useful building blocks for feature extraction system integration and more meaningful evaluation results.

The discussion in (2) makes it clear that even in an almost completely automated system, but where some human involvement is still required, the human-machine interface is a (possibly "the") critical component from the standpoint of application utility. An algorithm that makes errors that are easily fixed by the specific facilities of the available interface is more useful than a competing algorithm that makes fewer mistakes if these mistakes are not easily corrected by the available editing tools.

Thus, in order to determine their utility for a given application, algorithms must be evaluated in the context of both the task they will perform and the system in which they will be embedded. To achieve some generality in algorithm performance evaluation, it will be necessary to categorize the preconditions for algorithm invocation and the types of errors the algorithm makes. The algorithm can then be parameterized in terms of these categories, and its utility in a given application context computed as a weighted sum of its parametric characterization.

Because of the above considerations, it is desirable to have a somewhat larger number of narrow feature categories for classifying types of algorithms, rather than a few very broad categories that no single algorithm is likely to completely cover – i.e., we want the algorithms to completely cover a category rather than fall short of being able to deal with all the feature extraction problems that the category includes.

4. The purpose of a "Benchmark" type of evaluation should be validation – not discovery.

Even for a single site, it is extremely expensive in terms of both time and dollars to construct the reference models and collect the controlled datasets to perform a set of experiments; formal evaluation in the APGD program will be restricted to benchmark type experiments on a few selected sites.

We note that a benchmark is not a full statistical evaluation. We can't hope to use the benchmark experiments to discover the performance characteristics and range of applicability of feature extraction algorithms and systems, but must assume that this information will be provided by the designer – at best, the benchmark can be used as a validity check on designer provided performance information, and possibly, as a weak way of evaluating the relative performance of comparable algorithms with respect to previously untested conditions (e.g., the ability to use radar images in place of E.O. imagery).

We (SRI) argue that special information will often need to be collected if we try to benchmark an algorithm under conditions for which it is known that the algorithm was not designed to operate – or the numbers will have little meaning. Thus, for example, if an algorithm or system is designed to detect paved roads, but not dirt roads, it (the algorithm) might also have a stated requirement to operate on Radar imagery in which paved roads stand out from the background, but dirt roads are almost invisible. In a test in which both types of roads appear in the reference model, we know in advance that the algorithm will miss detecting all the dirt roads and will have wildly different completeness scores for images with different ratios of dirt to paved roads unless we define explicit object categories for dirt and paved roads. If a customer wants a system that can model both paved and unpaved roads, it will be necessary to employ EO imagery and additional algorithms that were designed to delineate dirt roads (or extract the dirt roads interactively in an editing operation). The key point in the above discussion is that while the evaluation must quantify the proportion of a given task that the algorithm (or system) is not able to deal with in any specific test context, it must also distinguish between algorithm failure and explicitly specified algorithm design restrictions or limitations.

Another important issue is that of algorithm "operating-point." Most algorithms can trade "missed detections" (false negatives) for false alarms (false positives) – the balance between these two types of errors defines what is called the operating point. Without an externally provided differential weighting, the algorithm designer (either implicitly or explicitly) picks a somewhat arbitrary operating point. Evaluating the relative performance of algorithms that have differently chosen operating-points can produce misleading results if the errors are later categorized and compared by category. It would be very desirable, but impractical, to ask the algorithm designer to provide a complete "operating characteristic curve" for his algorithm under a variety of different contextual conditions. However, it might be reasonable to get some indication of algorithm performance when either completeness or correctness is emphasized.

A central theme of the APGD effort is how to achieve modeling-system robustness and reliability. An algorithm that is robust and predictable under narrow but well 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 can't provide performance estimates or guidelines for its use in different contexts.

5. Precision of the Evaluation Process.

How accurate and repeatable is the evaluation. For example, if a second reference model is produced by a different analyst (or even by the original analyst at a later time), what is the

score one reference would be assigned with respect to the other. If this number is significantly different than 90-100%, the evaluation is not robust and has little value. It is important to recognize that the benchmark is a crude evaluation tool; assuming the computed evaluation scores have less than (say) a 10% variability would be highly optimistic. Therefore, trying to measure performance to better than a 10% degree of accuracy is merely adding noise (and cost) to the evaluation.

E APGD EVALUATION DATA FORMATS

E.1 Introduction

In order to facilitate as wide a community participation as possible in sharing of algorithms and results, we have adopted a very simple format for representing the geometry and topology of buildings and road networks. Where tradeoffs between generality and simplicity have been made, we have favored simplicity. Furthermore, these data models and formats are intended for communication of results for evaluation, *not* as a comprehensive exchange or storage format. In certain instances representations have been selected to make the evaluation task easier and in the process sacrificing generality. We expect to refine this format for subsequent evaluation or abandon it in favor of more comprehensive and general formats.

E.2 Syntax and File Format

This format is based on the syntactic conventions of Lisp and makes extensive use of a-lists to allow for easy parsing (using Lisp anyway) and to provide a somewhat “self-documenting” format. Zero-based indexing is used. Line breaks and indentation are not significant. All white space serves as a token separator. Semicolons not enclosed in double quotes cause the remainder of the line to be ignored by the reader.

All data files are divided into four parts: *tag*, *attributes*, *images*, and one or more *objects*.

E.2.1 Tag

All files begin with the string:

```
APGD-EVALUTION-FORMAT-V1.0
```

E.2.2 Attributes

The file attribute section is used to identify the file, test run, and so forth.

```
(FILE-ATTRIBUTES
 :author "Alice P. Grobner-Davis"
 :organization "Geospatial Models International"
 :email "grobner-davis@gmi.com"
 :date-time "4/29/98 02:40:23 PDT"
 :comment "GMI results Ft. Benning test area 1"
 :any-random-stuff "random stuff")
```

E.2.3 Images

The images section lists the images and corresponding camera parameters that were used for the extraction tasks. The images are specified by a URL. If an element of the image list is itself a list,

then the first element is interpreted as the image filename and the second as the camera-parameter file. If the base-url is specified, it is prepended to the image filename and camera-parameter filename.

```
(IMAGES
 :site "Ft. Benning"
 :base-url "http://www.ai.sri.com/~apgd/v1/datasets/Benning/panchromatic/"
 :image-list (("4_8.tif" "4_8.tec")
              ("4_7.tif" "4_7.tec")
              ("4_9.tif" "4-9.tec")))
```

E.2.4 Objects

After the introductory sections, one or more objects can be listed.

E.3 Primitives

E.3.1 Object Space Coordinates

Object space locations are specified in UTM (Universal Transverse Mercator) coordinates (in the site's zone: Ft. Benning is in UTM Zone 16; Ft. Hood is in UTM Zone 14) as a triple of double-precision float numbers in the order: easting (x), northing (y), elevation (z). The reference ellipsoid and horizontal datum are WGS84 (World Geodetic System 1984). The vertical datum is MSL (mean sea level). All lengths are measured in meters. As far as we know, this is consistent with current NIMA policy and practice.

E.3.2 Image Plane Coordinates

Image plane coordinate are given in pixel coordinates as a pair of (row column). These can be integers or floats.

In the case of integer coordinates, the origin, (0 0), is the upper-left-hand pixel in the raster. Row coordinates increase from top to bottom and column coordinates increase from left to right. If, for example and image has dimensions of 1024 rows and 1500 columns, the valid range for for row coordinates are integer values between 0 and 1023 inclusive and the valid range for column coordinates are the integer values between 0 and 1499 inclusive.

In the case of float coordinates the origin is taken as the upper-left-hand corner of the pixel in the upper-left-hand corner image and has the coordinates (0.0 0.0). Row coordinates increase from top to bottom and column coordinates increase from left to right. If, for example and image has dimensions of 1024 rows and 1500 columns, the valid range for for row coordinates is the semiclosed real interval $[0..1024)$ and the valid range for column coordinates is the semiclosed real interval $[0..1500)$.

E.3.3 Points

A point is a geometric primitive that specifies a position in object space. In addition, a point can carry information about the location in an image plane. The :position is given as a triple of coordinates in the current coordinate system (which, as described above will always be UTM for the time being). The optional :measurements field is specified as a list of triples, one per image in the order in which the images were listed in the IMAGES section. If a given vertex or point is not visible or was not measured in a particular image, an empty list "" or the symbol NIL is used as a placeholder. If no image measurements are given, the entire :measurements field can be omitted.

E.4 Road Network

A road network is a graph specified as a list of intersections (vertices) and a list of road segments (edges) connecting them. The network can contain one or more connected components. A road segment is an ordered list of sample points that lie along the centerline of the road. For the purpose display and evaluation, they are assumed to be connected by straight (in a local Cartesian system) line segments. Optionally, they may have an a-list of attributes.

Intersections have:

1. a position
2. a list of adjacent intersections
3. a list of the corresponding road-segments that connect to them
4. a list indicating which end of the given road-segment connects to the intersection.

E.5 Buildings

A building is a polygonal faceted object. Graphically, it is represented by a list of vertices and faces. Faces are specified as a list of zero-based vertex indices that define the perimeter of the face. Vertices are listed in clockwise order as viewed from the outside of the building.

To simplify evaluation, this description is further broken down into four components: cue-point, footprint, roof-faces, other-faces.

The cue-point is a single point anywhere within the footprint of the building and indicates the presence of a building. Roof-faces and other-faces may be specified, but for the current evaluation exercise, only a building's cue-point and footprint will be considered.

E.6 Complete Example

This section contains a simple, but complete, example of a road network and building model in the ASCII evaluation format.

```
1  APGD-EVALUATION-FORMAT-V1.0
2
3  (FILE-ATTRIBUTES
4    :AUTHOR "connolly"
5    :ORGANIZATION "SRI"
6    :DATASET-NAME "APGD Evaluation"
7    :EMAIL "connolly@ai.sri.com"
8    :DATE-TIME "Mon May 4 1998 17:21:54"
9    :COMMENT "Automatically generated by WRITE-APGD-EVALUATION-FILE.")
10
11
12  (IMAGES :SITE "Fort Benning 2"
13    :BASE-URL "http://www.ai.sri.com/~apgd/v1/datasets/Benning/panchromatic/"
14    :IMAGES (("4_8.tif" "4_8.tec")
15             ("4_7.tif" "4_7.tec")))
16
17
18  (ROAD-NETWORK
19    :ROADS
20    (
21      ;; Road Segment 0
22      (ROAD-SEGMENT
23        :POINTS
24        ((POINT :POSITION (706529.4512708816 3583616.522333523 129.99961275234819)
25              :MEASUREMENTS ((5049.056135292539 1893.9582749054642)
26                             (5125.357188700025 5390.6890623289955))))
27        (POINT :POSITION (706512.240544314 3583514.4268456837 127.58406674023718)
28              :MEASUREMENTS ((4908.0356352081135 1192.1213924813549)
29                             (4987.283414443198 4683.210735995194))))))
30      ;; Road Segment 1
31      (ROAD-SEGMENT
32        :POINTS
33        ((POINT :POSITION (706582.2590351252 3583609.8305805805 129.70875930693)
34              :MEASUREMENTS ((5416.000132355148 1835.0645055925475)
35                             (5491.615911369796 5335.29865518779))))
36        (POINT :POSITION (706543.6743908097 3583618.021911892 130.0000002803281)
37              :MEASUREMENTS ((5148.533212343847 1900.9812123032982)
38                             (5224.589596707213 5398.784575469816))))
39        (POINT :POSITION (706529.4512708816 3583616.522333523 129.99961275234819)
40              :MEASUREMENTS ((5049.056135292539 1893.9582749054642)
41                             (5125.357188700025 5390.6890623289955))))))
42      ;; Road Segment 2
43      (ROAD-SEGMENT
44        :POINTS
45        ((POINT :POSITION (706457.2928798852 3583601.499992322 129.15437119826675)
46              :MEASUREMENTS ((4543.591598993336 1808.9586866060445)
47                             (4620.8950190301985 5297.262666888984))))
48        (POINT :POSITION (706516.3775357847 3583617.447717102 130.00000027753413)
```

```

49         :MEASUREMENTS ((4958.109891512203 1903.608008961294)
50             (5034.546710916542 5399.233641693385)))
51     (POINT :POSITION (706529.4512708816 3583616.522333523 129.99961275234819)
52         :MEASUREMENTS ((5049.056135292539 1893.9582749054642)
53             (5125.357188700025 5390.6890623289955))))))
54
55 :INTERSECTIONS
56 (
57     ;; I-0
58     (INTERSECTION :POSITION
59         (POINT :POSITION (706457.2928798852 3583601.499992322 129.15437119826675)
60             :MEASUREMENTS ((4543.591598993336 1808.9586866060445)
61                 (4620.8950190301985 5297.262666888984)))
62         :ADJACENT-INTERSECTIONS (1)
63         :INCIDENT-ROADS (2)
64         :INCIDENT-ROAD-DIRECTIONS (HEAD))
65     ;; I-1
66     (INTERSECTION :POSITION
67         (POINT :POSITION (706529.4512708816 3583616.522333523 129.99961275234819)
68             :MEASUREMENTS ((5049.056135292539 1893.9582749054637)
69                 (5125.357188700025 5390.6890623289955)))
70         :ADJACENT-INTERSECTIONS (0 2 3)
71         :INCIDENT-ROADS (2 1 0)
72         :INCIDENT-ROAD-DIRECTIONS (TAIL TAIL HEAD))
73     ;; I-2
74     (INTERSECTION :POSITION
75         (POINT :POSITION (706582.2590351252 3583609.8305805805 129.70875930693)
76             :MEASUREMENTS ((5416.000132355148 1835.0645055925475)
77                 (5491.615911369796 5335.29865518779)))
78         :ADJACENT-INTERSECTIONS (1)
79         :INCIDENT-ROADS (1)
80         :INCIDENT-ROAD-DIRECTIONS (HEAD))
81     ;; I-3
82     (INTERSECTION :POSITION
83         (POINT :POSITION (706512.240544314 3583514.4268456837 127.58406674023718)
84             :MEASUREMENTS ((4908.0356352081135 1192.1213924813549)
85                 (4987.283414443198 4683.210735995194)))
86         :ADJACENT-INTERSECTIONS (1)
87         :INCIDENT-ROADS (0)
88         :INCIDENT-ROAD-DIRECTIONS (TAIL))))))
89
90 (BUILDING
91     :CUE-POINT
92     (POINT :POSITION (706541.7159124829 3583603.1354591036 135.6997930817306)
93         :MEASUREMENTS ((2568.4151605271964 890.0611276730099)
94             (5213.230663310671 5302.3105265372815)))
95     :POINTS
96     ((POINT :POSITION (706531.3664051673 3583597.0150699145 130.03319429792464)
97         :MEASUREMENTS ((2529.814825367743 877.9822582530535)
98             (5135.504820593029 5255.022940566763)))
99     (POINT :POSITION (706549.4623846987 3583593.9484108603 130.01908788178116)
100        :MEASUREMENTS ((2592.6358047450977 865.0463086667168)
101            (5260.979260094253 5230.746900740439)))
102     (POINT :POSITION (706552.056286842 3583609.254731569 130.01426505856216)

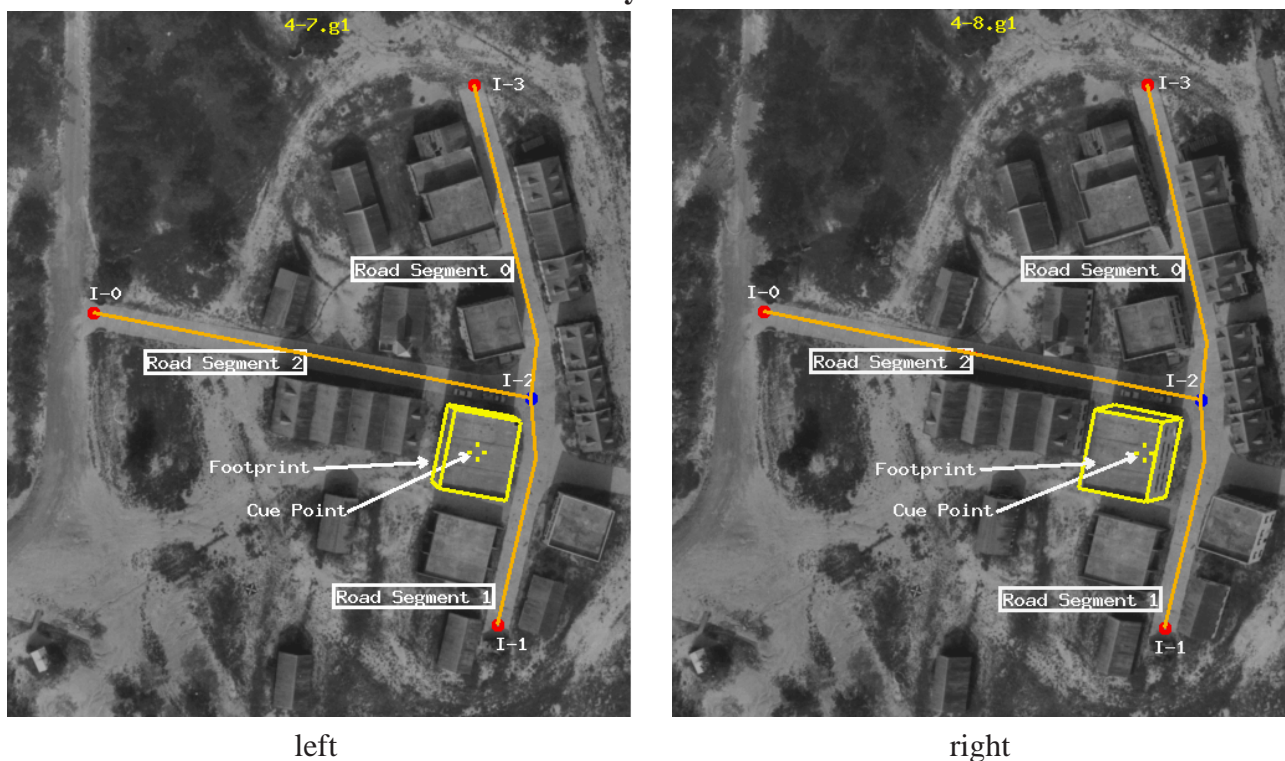
```

```

103      :MEASUREMENTS  ((2603.1629822443138 918.2862668201107)
104      (5281.525011353987 5336.588497799024)))
105 (POINT :POSITION (706533.9603068119 3583612.3213892216 130.02837151288986)
106      :MEASUREMENTS  ((2540.347793437586 931.2023594812042)
107      (5156.068289660162 5360.8391496884515)))
108 (POINT :POSITION (706531.3755568907 3583597.016198233 141.38534374162555)
109      :MEASUREMENTS  ((2533.218970995635 861.4717860140258)
110      (5144.045237461665 5267.585309781727)))
111 (POINT :POSITION (706549.4715042469 3583593.9495446608 141.37123735249043)
112      :MEASUREMENTS  ((2596.8639844262944 848.360755316517)
113      (5271.162189387799 5242.994367291143)))
114 (POINT :POSITION (706552.0654017777 3583609.2558380235 141.36641453392804)
115      :MEASUREMENTS  ((2607.528396403898 902.3004946835558)
116      (5291.974767440065 5350.217612050879)))
117 (POINT :POSITION (706533.9694539227 3583612.3224901934 141.3805209621787)
118      :MEASUREMENTS  ((2543.889326596661 915.3911433089036)
119      (5164.87600031842 5374.78249934925))))
120 :FOOTPRINT (0 3 2 1)
121 :ROOF-FACES ((5 6 7 4))
122 :OTHER-FACES ((0 4 7 3) (2 6 5 1) (0 1 5 4) (2 3 7 6))
123 )

```

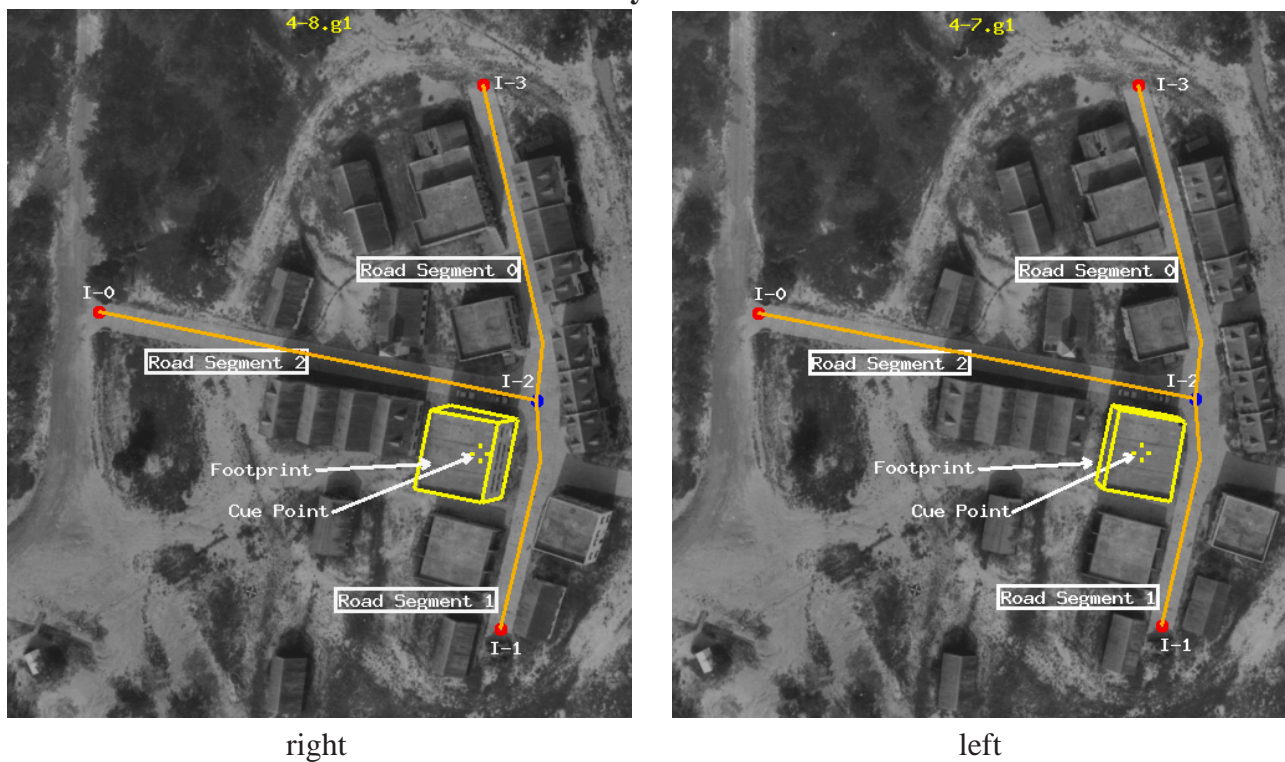
“Wall-eyed” stereo



left

right

“Cross-eyed” stereo



right

left

Figure 16: A stereo-pair of images showing the extracted road network and building. The labels correspond to sections of the ASCII evaluation format that follows.

F VIDEOTAPE OF FIRST ANNUAL DEMONSTRATION

A videotape that briefly describes the first annual demonstration is part of this report.