

# On the Evaluation of APGD Algorithms and Systems

M. A. Fischler, A. J. Heller, and R. C. Bolles  
Artificial Intelligence Center  
SRI International, Menlo Park, CA 94025

May 4, 1998

## Preface

*This document is a first attempt to reach 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.*

*Evaluation Day at Terrain Week (EDTW98)<sup>1</sup> will offer an opportunity for community-wide discussion and modification of the enclosed material. In fact, we and our government sponsors view the main goal of the meeting as evaluating the evaluation process, not the algorithms and systems themselves. It is expected that DARPA-sponsored APGD researchers attending the meeting will participate in the road and building extraction experiments. Their results can be self-evaluated or evaluated over the WWW using SRI-developed software and facilities. A discussion of the results of such experiments can serve as the basis for suggested changes in the procedures proposed in the next section of this document.*

*There is a discussion of the rationale and assumptions underlying our proposed evaluation metrics, as well as some of the more philosophical issues in Appendix 4, which is entitled "APGD Evaluation Philosophy and Rationale."*

*Appendix B contains explicit specifications for the data models and ASCII file formats for roads and buildings.*

---

<sup>1</sup>18 May 1998 in Austin, TX.

## Contents

<b>1</b>	<b>Proposed APGD Evaluation Metrics</b>	<b>3</b>
<b>2</b>	<b>Report Format</b>	<b>5</b>
<b>3</b>	<b>The Road Evaluation Process</b>	<b>7</b>
3.1	Segment Geometry . . . . .	8
3.2	Segment Attributes . . . . .	8
3.3	Network Topology . . . . .	8
<b>4</b>	<b>The Building Evaluation Process</b>	<b>9</b>
<b>A</b>	<b>APGD Evaluation Philosophy and Rationale</b>	<b>10</b>
A.1	Discussion of Critical Issues and Assumptions . . . . .	11
<b>B</b>	<b>APGD Evaluation Data Formats</b>	<b>18</b>
B.1	Introduction . . . . .	18
B.2	Syntax and File Format . . . . .	18
B.2.1	Tag . . . . .	18
B.2.2	Attributes . . . . .	18
B.2.3	Images . . . . .	19
B.2.4	Objects . . . . .	19
B.3	Primitives . . . . .	19
B.3.1	Object Space Coordinates . . . . .	19
B.3.2	Image Plane Coordinates . . . . .	19
B.3.3	Points . . . . .	20
B.4	Road Network . . . . .	20
B.5	Buildings . . . . .	20
B.6	Complete Example . . . . .	22
<b>C</b>	<b>General References</b>	<b>26</b>

# 1 Proposed APGD Evaluation Metrics

Components of the derived model will be compared with those of the provided reference model and will be evaluated with respect to the following basic metrics (plus additional metrics that may be appropriate for specific features or applications). It should be noted that 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.

Multiple tests may be run depending on the purpose of the evaluation (see forthcoming document on Rationale and Assumptions).

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

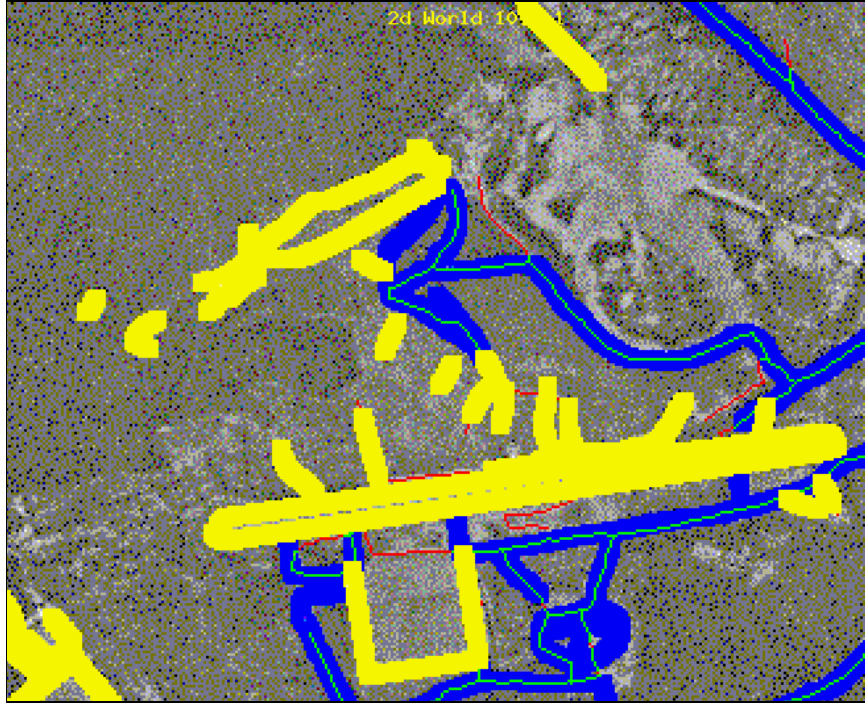
**Reference Model** An object space model generally recognized as representing the “correct” answer for the feature extraction task under evaluation.

**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 CARE	DC	Included in the Reference model in the DON'T CARE class, but not considered to be either correct (TP) or incorrect (FN) when included or not included in the Derived model.

We foresee two reasons for using a DON'T CARE value. First, if cartographers cannot agree on whether or not an item should be included in the reference data, we plan to mark the item as a DON'T CARE. Second, if we want to run a specialized test, we might mark entire classes of items as DON'T CARE for that one test. For example, if we have a technique to detect dirt roads, but not paved roads, we might temporarily set all paved roads in the reference model to DON'T CARE, and then test our algorithm on the unpaved roads in the reference model. The issue of the use of the DON'T CARE value will be discussed in more detail in a later document concerned with evaluation rationale.



**Figure 1:** An example of the scoring of a road network extraction result. The TRUE POSITIVE reference model is shown in blue. The DON'T CARE reference model is shown in yellow. Roads scored as TRUE POSITIVE are shown in green. Roads scored as FALSE POSITIVE are shown in red.

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 “detection percentage:”

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

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

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

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

This metric can vary from 0 to infinity (a small 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.

$$100 \times \frac{TP}{(TP + FP)}. \quad (3)$$

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

**Robust Correctness** (at the 90% confidence level): a measure of the algorithm's ability to evaluate its own output and only return correct instances of the object class for which it is searching.

$$100 \times \frac{TP - 9FP}{TP + FN}. \quad (4)$$

The best possible score is 100; a score of zero or greater is a very good result (i.e., fewer than one FALSE POSITIVE for each nine TRUE POSITIVE instances); the worse possible score is minus infinity.

## 2 Report Format for Describing an Extraction Algorithm and the Results of Applying It to a Dataset

This is the information a researcher is encouraged/expected 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 of the derived models is required on an orthophoto or in 3-D in a stereo model.

### 3 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.

### 3.1 Segment Geometry

The procedure for evaluating centerlines at the sample-point level involves projecting the centerline points into the provided orthoimage, if they are not stated in orthoimage coordinates, and comparing them pixel for pixel with the 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 miles of road.

### 3.2 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 plan to estimate a metric that we call the average-connectivity-interruptions per mile. 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 mile.

### 3.3 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.

## **4 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 B.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 image 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.

## A 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 only be compared against paved road reference data. For this type of evaluation, features may be labeled "Don't Care" in the reference model if they are not relevant to the problem at hand or within the intended scope of the algorithm's competence.

In a benchmark exercise (such as the one we are planning for Ft. Hood), 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) assign a time to 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.

(Note that approaches 1 and 2 are extremes. Approach 3 falls in-between.)

## **A.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 care" 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 his 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-cares(DC)” 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 performance. 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. For practical reasons, in the Terrain-Week98 exercise , we will employ somewhat arbitrary fixed (but reasonable) tolerances on the derived model.

For roads, we will accept as correct any centerline that is within 1-2 road-widths of the reference-model centerline (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, and there may be some quantization error due to our evaluation technique, and there might also be some projection error if the image-analyst extracted the roads in 3-D using a stereoscope).

For buildings, which are represented by the 3-D location of the vertical-wall and planar roof surface intersections, we will accept a 6-7 pixel x-y deviation in the highest resolution (stereo-pair) imagery provided. The corresponding object-space height (z) tolerance can be computed by mapping the allowed image-error back through the provided camera model. The rationale for choosing these tolerances is based on the observation that semi-automatic feature extraction techniques (e.g. RCDE and MBO systems) can typically locate "local" building 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 15-30 cm "X-Y" error and a 1 meter error in "Z". 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 image space (and corresponding distances in object space), which roughly corresponds to a just noticeable difference to the eye when the model is superimposed on the image.

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.

Our plans for Terrain-Week98 are focused on the automatic extraction problem and do not include provision for an editing step but this issue will be addressed in future evaluations.

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; in the latter case, the algorithm is not penalized for not finding the dirt roads (just like it should not be penalized for not finding any buildings that might also be present). 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.

We (SRI) have proposed a metric called robust correctness to provide an indication of the ability of an algorithm to perform self-evaluation of its results and only return valid instances of the feature class it is searching for (i.e., emphasizing correctness over completeness). This does not imply that we have no interest in other operating points of the algorithms being tested.

#### 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 100%, the evaluation is not robust and has little value. Similarly, if what appears to be some minor change in scene content causes a large change in the score an algorithm receives, the evaluation has little value. It is important to recognize that the benchmark is a crude evaluation tool; assuming the computed 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.

#### 6. Terminology

We will never be able to assign a set of names to our chosen metrics that the APGD community will unanimously agree upon. It is not worth the time to argue over such names if they are reasonably self-descriptive and are well defined – especially if they provide some historical continuity with past usage.

## B APGD Evaluation Data Formats

### B.1 Introduction

In order to facilitate as wide a participation as possible for this exercise, 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.

### B.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*.

#### B.2.1 Tag

All files begin with the string:

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

#### B.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")
```

### B.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")))
```

### B.2.4 Objects

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

## B.3 Primitives

### B.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.

### B.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 an image has dimensions of 1024 rows and 1500 columns, the valid range for row coordinates is the semiclosed real interval [0..1024) and the valid range for column coordinates is the semiclosed real interval [0..1500).

### **B.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.

### **B.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.

### **B.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.

## B.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
```

“Wall-eyed” stereo



left

right

“Cross-eyed” stereo



right

left

**Figure 2:** A stereo-pair of images showing the extracted road network and building. The labels correspond to sections of the ASCII evaluation format that follows.

```

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 )

```

## **C General References**

TEC-SR-7, Handbook for Transformation of Datums, Projection, Grids and Common Coordinate Systems, US Army Corps of Engineers, Topographic Engineering Center, Ft. Belvoir VA.