

The Site-Model Construction Component of the RADIUS Testbed System

A. Heller, P. Fua, and C. Connolly

Artificial Intelligence Center, SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
E-MAIL: {heller,fua,connolly}@ai.sri.com

J. Sargent

BDM International
1501 BDM Way
McLean, VA 22102
E-MAIL: jsargent@bdm.com

Abstract

The central component in the RADIUS model-supported image exploitation paradigm, is the creation of a 3-dimensional model that captures the basic geometry of the site under examination. While many fully automated methods for site model construction show promising results, none are robust or general enough to replace interactive and semi-automatic techniques. In this paper, we outline the suite of site-model construction tools being assembled for the RADIUS program, a sample scenario for their use, and the results of a study to characterize the potential saving of effort they offer.

1 Introduction

The construction and use of 3-D models of military and industrial sites will allow revolutionary advances in the speed, confidence, and range of analytical techniques with which an image analyst develops and reports intelligence information.

The central component of this *model-supported image exploitation* paradigm, is the creation of a 3-dimensional model that captures the basic geometry of the site under examination. As part of the RA-

DIUS program, SRI has developed and assembled a suite of manual and semi-automatic tools for site-model construction.²

Figure 1 shows a typical RADIUS site model. The majority of the features modeled fall under three broad categories:

- Buildings and other structures such as petroleum and water storage tanks.
- Lines of Communication such as roads, railroad tracks, and other linear features such as rivers and streams.
- Functional Areas such as parking lots, site perimeters, rail transfer points and other area features such as forested areas.

Both manual and semi-automatic techniques for modeling each of these types of features are available in the RADIUS Testbed System (RTS).

Fully automatic techniques for modeling buildings and roads are under development as part of the RADIUS project and some are currently being evaluated for inclusion in the RADIUS Testbed System. While many show promising results, none are robust or general enough to do the full task on their

¹This work was sponsored by the the Advanced Research Projects Agency under contract DACA76-92-C-034 monitored by the U.S. Army Topographic Engineering Center, Ft. Belvoir, MD and by Lockheed-Martin Corporation under subcontract RRM881032. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Advanced Research Projects Agency, the United States Government, Lockheed-Martin Corporation, or SRI International.

²In this context, we define manual techniques as those in which the 3-dimensional model of a feature is projected into one or more images and the operator adjusts the model to align it with what is seen in the images; and semi-automatic techniques as those which require an operator to provide an initial rough estimate of a feature's position, size and topology and the system then refines or extends the model of the object using information extracted from the image(s).



Figure 1: A typical RADIUS site model, projected onto an image of the site. The site model mainly consists of buildings, linear and area features.

own. The most competent of these still require editing of the result, so even if automatic techniques are eventually integrated into the system, manual and semi-automatic tools will remain a necessary component.

2 The Site Model Construction Process

2.1 Perimeters and Functional Areas

Perimeters and functional areas (e.g., reporting positions, vehicle parks, forested areas, rivers and other natural features) can be manually or semi-automatically defined. Such areas can be used to control which regions in the image are suitable for automatic or semi-automatic model construction. This allows the model builder to guide automatic

or semi-automatic site modeling tools by restricting their regions of application. Such regions can be initially selected by marking points in an image along the boundaries of the desired regions. The RADIUS Testbed System displays these boundaries as they are being constructed on the image, making the results immediately available for review. The model builder can then adjust the boundaries of these regions to better fit the results to the images.

Area Features are modeled in the Radius Testbed System using the *3d Closed Curve* primitive. This is simply a sequence of 3-D vertices that are connected with either straight line segments or fit to a spline curve.

After positioning the initial vertex, the *Add Vertex* operation is used to interactively sketch the shape of

the feature. At this point the semi-automatic Model-Based Optimization (MBO) system [Fua, 1995] can be invoked to refine the fit of the model to the shape of feature in the image. Figure 2, shows the model of a pond before and after optimization. In this example, the operator entered four vertices and the MBO system inserted another 12 vertices to closely model the shape seen in the image. Alternatively, if a polygonally shaped area, such a parking lot, is being modeled, the user can specify this to the MBO system and no new vertices will be added during the optimization.

2.2 Lines of Communication (LOC)

The first step in semi-automatically modeling Lines of Communication (LOCs) such as roads, rail lines, and runways is to select an appropriate width. For example, primary roads and rail lines are typically wider than secondary roads. The model builder can then select a tangent direction and/or some points along the LOC to provide approximate position, orientation, length and width of the LOC in the image. The RADIUS Testbed System can then adjust the shape, placement, orientation and size of the feature model to agree with the image. The results of the semi-automated site model construction are continuously displayed to the model builder. The model builder can either interact with the system during this adjustment phase, or compare the final results in multiple images. The model builder can then make adjustments of unacceptable results using manual or other semi-automatic site modeling processes.

Roads are modeled using the *ribbon* primitive. This is a 3-D curve that has an additional parameter at each vertex to specify the width of the curve at that point. Two tools are currently available in the RTS to model LOCs: the SRI Road Tracker [Quam, 1978] and Ziplock Snakes [Neuenschwander *et al.*, 1994].

To use the Road Tracker, first an initial segment of the road is modeled. Then the *extend* function is invoked to model the remainder. The road tracker will stop at discontinuities in the photometry of the image of the road or possibly take the wrong path in ambiguous situations, such as intersections. In this case, manual techniques are used to edit the result. After the correction, the tracker can be reinvoked.

The ziplock snake is initialized by quickly dropping two or more critical points along the road being modeled. These are points through which the centerline of the finished ribbon is guaranteed to pass. When the ziplock snake optimization is invoked, successive segments of the road are clamped to the image of the road, similar fashion the process of closing a ziplock plastic bag. The result is a ribbon that closely follows the road. This tool is often easier to control than the Road Tracker, because the full extent of the road to be modeled is specified in advance.

Finally, a road that has been initially modeled either fully manually or with the road tracker or ziplock snakes can be simultaneously optimized in two or more images with the MBO system, resulting in a highly accurate, full 3-D model of the road. Figure 3 shows the pre- and post- optimization versions of a road.

2.3 Buildings and Structures

To model buildings and structures, the model builder either adds new structures or copies and adjusts structures already in the site model. To add a structure, the model builder can select from a number of primitive shapes in a library. A shape is selected to approximate a structure found in the image, e.g., cylinders for towers, or rectangular wireframes for buildings. The primitive shape is placed at the approximate position and orientation of the image feature to be modeled. The RADIUS Testbed System then adjusts the shape, placement, orientation, and size of the primitive to better fit the image.

Most buildings are modeled using the following primitives:

- *Boxes* which are simple rectangular solids.
- *Houses* which are boxes with sloped roofs. The system currently supports three kinds of roofs: gable, shed, or hip. The slope of the roof can be adjusted as well as the amount of overhang.
- *Quonset Huts* which are boxes with a half cylinder roof.
- *Extrusions* which are a curve that typically defines the (flat) roof line which is extended to the ground to form the sides of the building

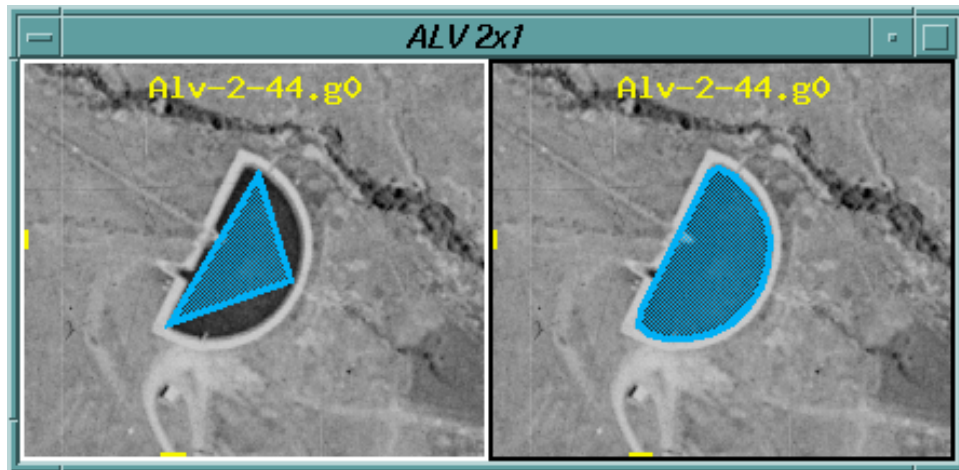


Figure 2: An area feature before and after optimization.

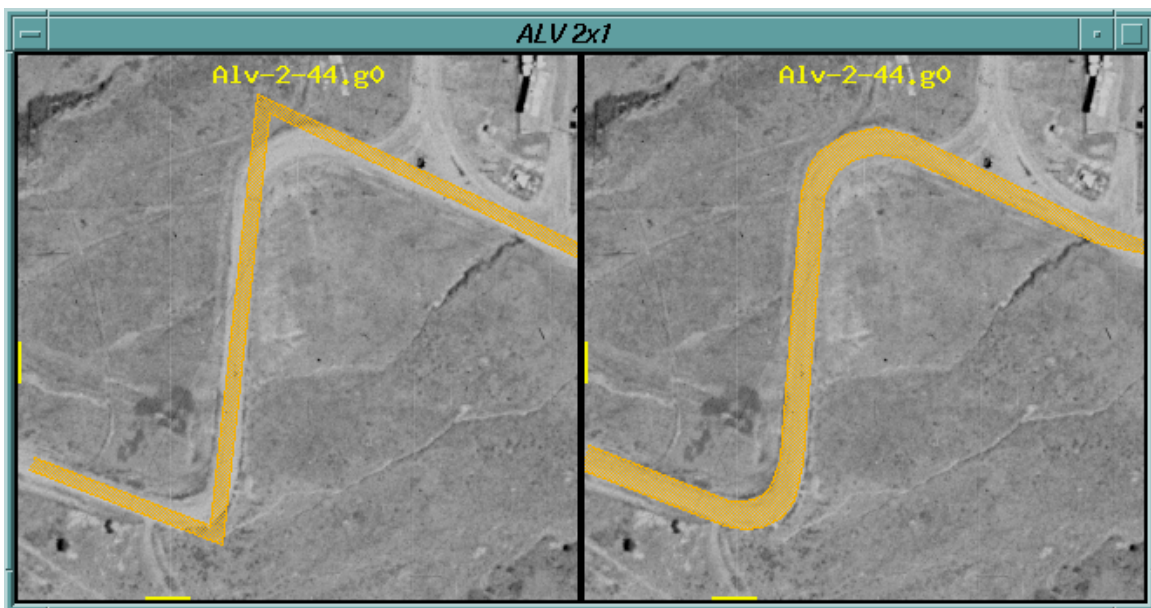


Figure 3: A road model, before and after optimization.

- *Composite Structures* which are a grouping of other primitives.

In addition, generalized superquadric and superellipse primitives are supported to model more rounded structures such as water towers.

2.3.1 Modeling a Building with a Box Primitive

After the primitive is added to the scene, the user grabs an upper corner of the object and adjusts the pose and size of the box to match the building being modeled with this sequence of operations:

1. *Move XY* to align the one corner of the box with a roof corner of the building being modeled.
2. *Rotate-Scale* to align the major and minor axes of the box primitive with those of the building.
3. *XY-Size* to adjust the length and width of the box.

At this point the top of the box is aligned with the roof of the building in a single image. The building height is set in one of two ways. If an accurate digital terrain model has been incorporated in to the

site model, the *Bury Foundation* operation will drop the bottom vertices of the box so that they are all underground. Alternatively, the height of the roof line can be triangulated, by examining how the box object overlays in two or more image. The *Move W* operation moves the box along the camera ray of the selected image and along the corresponding epipolar lines in the the other images. Finally, the height can be set, using the *Z-Size* operation to adjust the base of the box to match the bottom of the building as seen in the images. Figure 4 shows the progression of this operation.

Modeling a building with a *house* primitive follows the same sequence of steps, followed by setting the roof type (gabled, hip, shed), pitch, and overhang.

2.3.2 Modeling a Building with an Extrusion Primitive

Extrusions are used to model more complex shaped buildings. First the roof line of the building is sketched using a closed-curve object, by adding vertices to the curve using the *Add Vertex* operation. Then the height of the curve is triangulated using the *Move Curve W* operation, and finally it is extruded to the ground using the *Bury Foundation* or *Z-Size* operations.

2.3.3 Using the MBO system on structures

In the construction procedures for both simple box objects and extrusions, semi-automatic assistance can be invoked by a user using the Model-Based Optimization system. This system can be used to both automatically triangulate the elevation of the structure and then refine its shape to the features in the image. Since these last two are the most time consuming part of modeling a building, the Model-Based Optimization system offers a considerable enhancement in modeling productivity. Figure 5 shows the sequence of operations used in modeling a complex-shaped building with an extrusion primitive and the MBO system.

3 Characterization

One of the major goals of this work is to increase the speed and accuracy with which site models can be constructed from current imagery by developing a

new family of image understanding (IU) techniques. An obvious question to ask is just how well have we accomplished this.

We have chosen to measure the amount of effort expanded by the human analyst by the number of mouse-clicks and the amount of mouse-travel required to achieve a desired answer. We feel that this is a better measure than, for example, actual computation times because it truly reflects the amount of human interaction and does not depend on the speed of the computer being used.

We first briefly describe the code instrumentation that was required to perform the experiments and then report our results. For additional details we refer the interested reader to our quarterly reports.³

3.1 Instrumenting the RTS

We have developed and installed code that captures low level information from the RTS user interface about individual actions taken by the analyst. Every mouse motion associated with making adjustments to object parameters, and every mouse click is captured into an event history. Below is a list of the information being recorded:

- Object Adjustment Events:
 - Object ID
 - Event start time
 - Adjustment type: (for example: vertex-xy, vertex-z, vertex-width, ...)
 - 2-D world ID
 - Zoom level
 - Sequence of time deltas and mouse-position deltas of the form: (delta-t dx dy)
- Mouse Click Events:
 - Object ID
 - Event start time
 - Event id (for example: zoom-in, zoom-out, recenter, drop-z)
 - 2d-world ID

³These can be retrieved via the URL: <http://www.ai.sri.com/radius/sri/baa-reports/>.

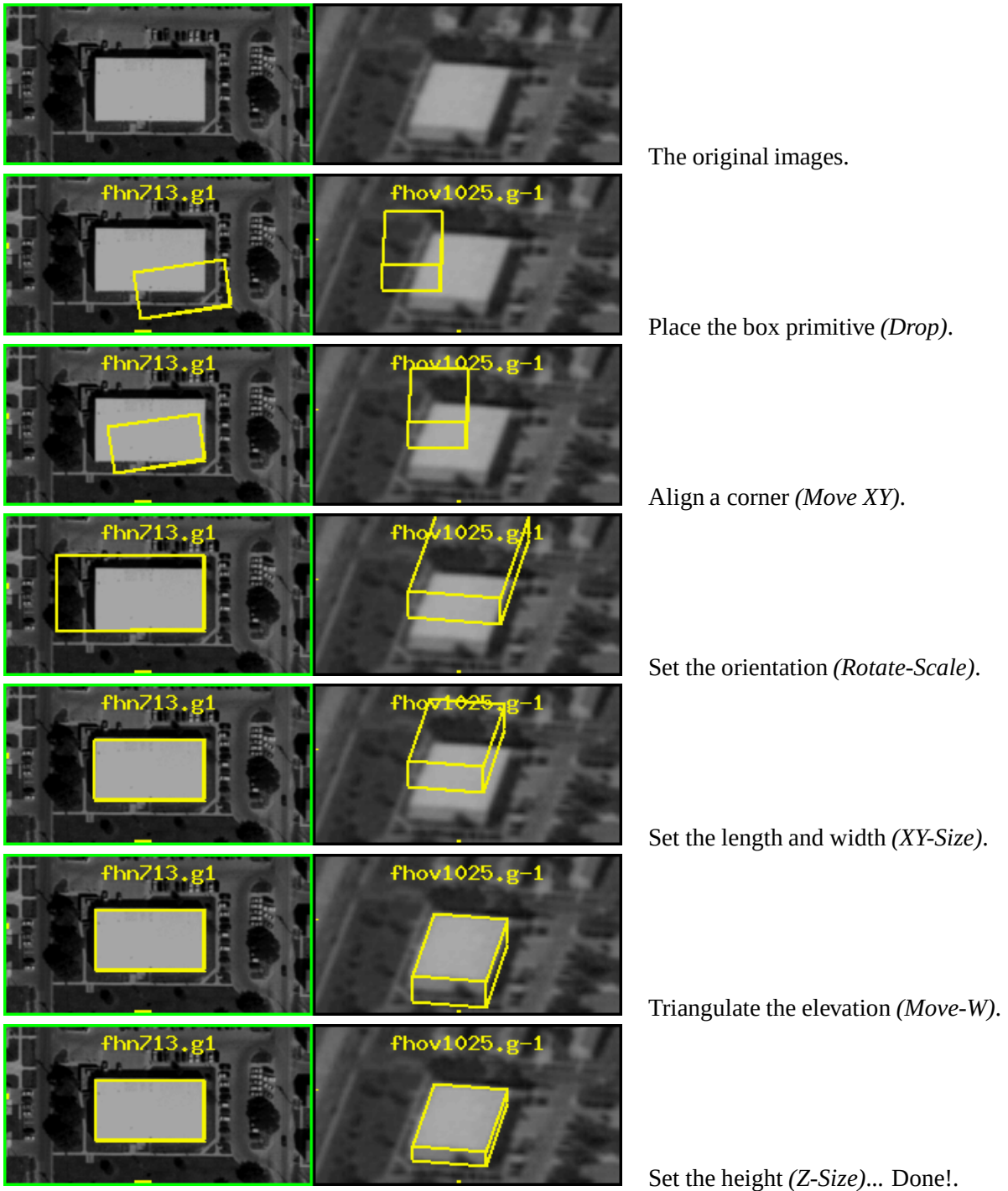


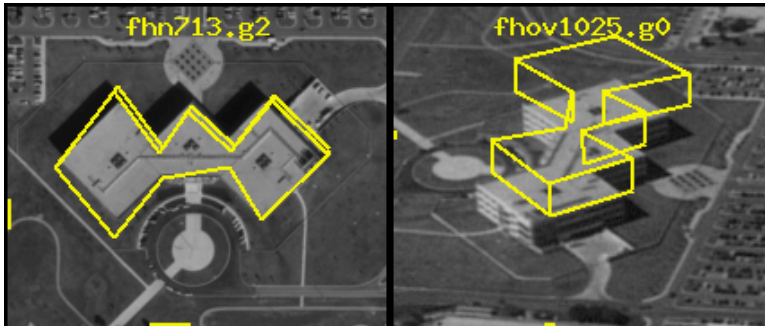
Figure 4: The sequence of adjustments used to manually model a building.



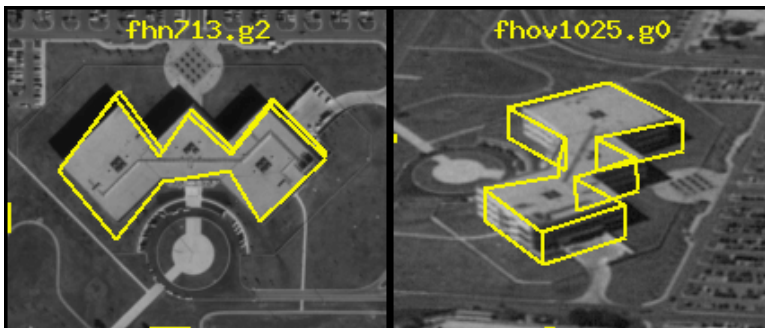
The original images.



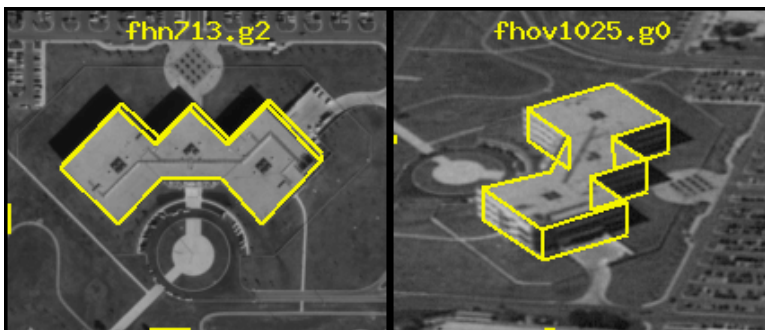
Sketch roof-line (*Add Vertex*).



Done with roof-line (*Drop*).



Find elevation (*MBO-Z-Search*).



Optimize Shape (*MBO-Opt*)... Done!

Figure 5: The sequence of steps used to model a complex-shaped building with the extrusion primitive and the Model-Based Optimization system. This entire sequence typically takes less than 1 minute of elapsed time.

- Zoom level
- Mouse 2-D world position

This event history then summarized by a small number of meaningful numbers. Among them are:

- Number of mouse clicks
- Number of mouse moves
- Total distance mouse moved during adjustments
- Total time in adjustment events
- Total time in fine adjustments

We have also implemented a metric to estimate the precision of an extracted road by comparing the centerline of the extracted road to the ground truth centerline. For each vertex of the extracted road, the distance to the nearest centerline point of the ground truth centerline is computed. The data is reduced to the following two numbers:

- Mean vertex to ground truth distance
- Maximum vertex to ground truth distance

These seven numbers appear in the figures of the following section.

3.2 Experimental Results

We used images such as the 7000×7000 example shown in Figure 6 to perform our experiments and chose a set of road segments to be modeled as accurately as possible. We chose four approaches that are representative of the methods used to delineate roads using the RTS. They are:

HAND – Hand-tracing using the RTS interface but neither snakes nor road tracker.

TRACKER – Using SRI’s road tracker [Quam, 1978] to provide the initial sketch in the full resolution image and then refining it using a ribbon snake [Fua, 1995].

SNAKE1 – Sketching the road using the full resolution image and refining it using a ribbon snake.

SNAKE2 – Sketching the road using a half-resolution version of the image, refining it using a ribbon snake first at half-resolution and then at full resolution.

In all cases we used the system’s default parameter settings and allowed the user to manually refine the automatically generated results to produce satisfactory delineations. The bar graphs that appear in the Figures 7 and 8 are labeled HAND, TRACKER, SNAKE1, SNAKE2.

We used the hand-traced versions of the roads as our references and the metric discussed above to evaluate the quality of the delineations produced by the three semiautomated approaches. As shown in Figure 7, the results are virtually indistinguishable in terms of average distance whereas the TRACKER approach does better in terms of maximum distance.

Figure 8 depicts the amount of effort required by each approach, as measured by the number of mouse clicks and the amount of mouse travel. The TRACKER approach appears to be very effective and yields at least a sixfold improvement on all counts except the total number of mouse clicks. This is due to the fact that starting and stopping each operation—automated sketching and snake refinement—require several clicks. This number could be drastically reduced by defining user-interface methods that sequentially perform all these operations with a single mouse-click.

The SNAKE2 approach is almost as effective but requires more effort to provide the initial sketch. This problem could be alleviated by using ziplock snakes [Neuenschwander *et al.*, 1994] instead of traditional snakes.

For all three semiautomated methods, however, a large portion of the human interaction goes into specifying the width of the road as the current tools have no way of computing it. Therefore, methods to compute the width of a road given only its centerline would be extremely valuable and should be the object of future research.

In short by further improving the interface and developing a width-computing algorithms, we should be able to turn the current six-fold reduction of effort into a ten- to hundred-fold one.



Figure 6: An image with two overlaid roads.

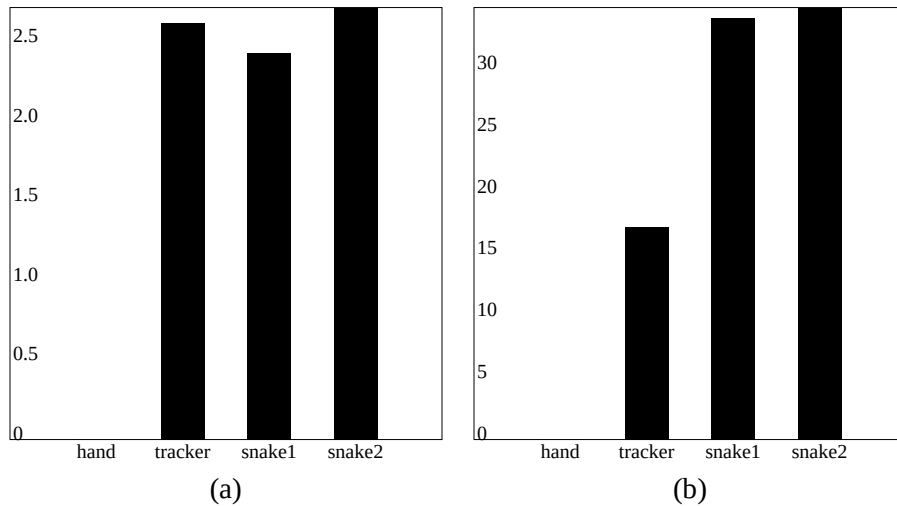


Figure 7: Distance to hand-entered roads. (a) Average distance difference (b) Maximum distance difference. Because the hand-entered results are taken to be the reference, the corresponding distances are zero.

4 Conclusions and Future Work

We have described the manual and semi-automatic facilities available in the RADIUS Testbed System for modeling common buildings and other structures, and linear and area features found at sites of intelligence interest, and have presented a sample

scenario of their use. We also report on an experiment that shows that a six-fold reduction in effort of delineating linear structures, such as roads, is possible by using the Image Understanding-based semi-automatic facilities available in the RTS.

The current image understanding-based tools are

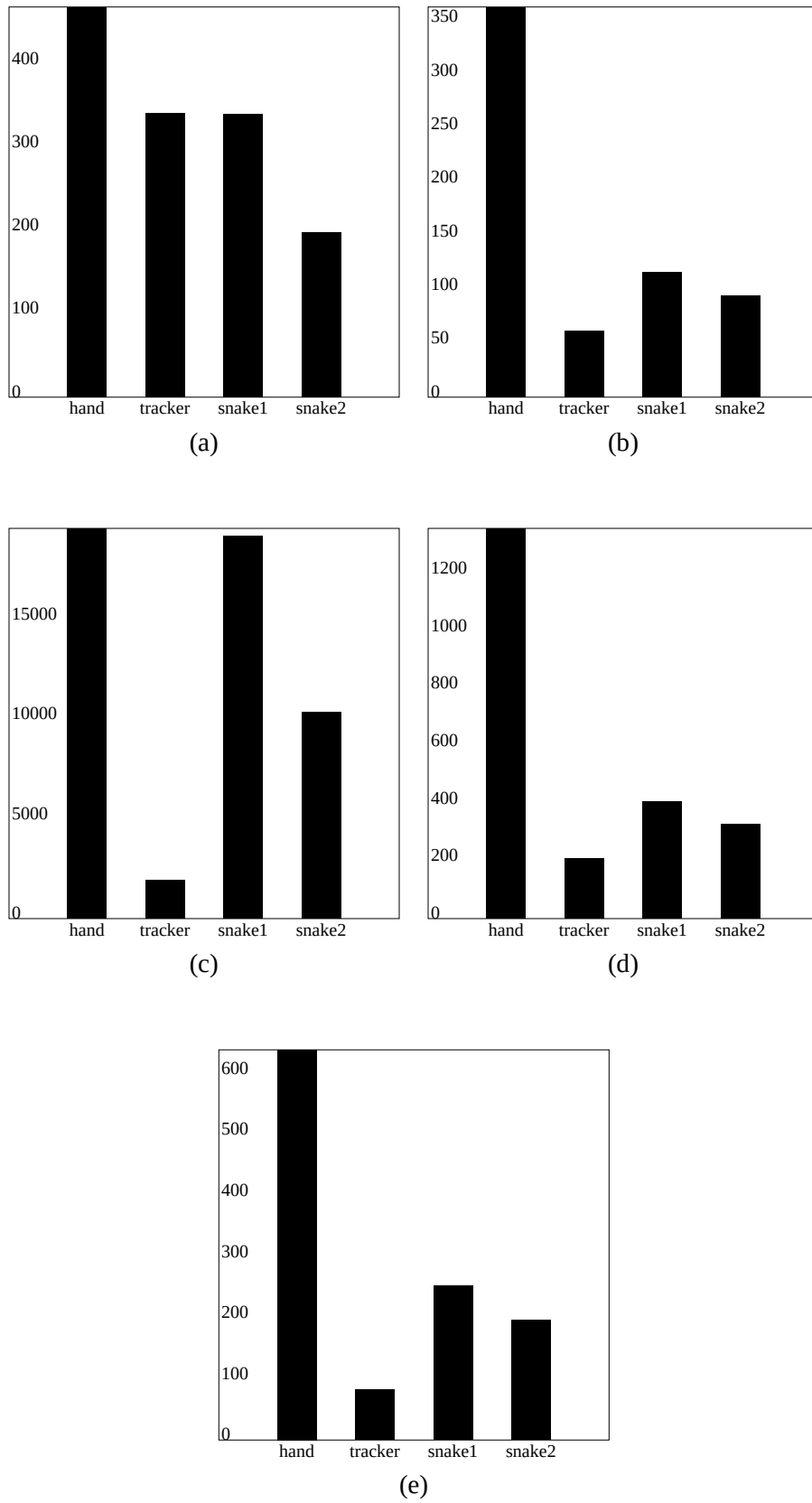


Figure 8: Amount of effort. (a) Number of object clicks. (b) Number of mouse moves. (c) Total mouse distance. (d) Total mouse move time. (e) Total near mouse move time.

largely based on the Model-Based Optimization approach, which in its current implementation, uses fairly simple objective functions that are mainly based on image gradient strength. At this point in time, these do not employ domain-specific or contextual information in the optimization process. The ability to incorporate information such as road and building construction practices, or to impose inter-object constraints such as coplanarity or adjacency of faces may potentially yield even greater improvements in productivity.

References

- [Fua, 1995] P. Fua. Parametric Models are Versatile: The Case of Model Based Optimization. In *ISPRS WG III/2 Joint Workshop*, Stockholm, Sweden, September 1995.
- [Neuenschwander *et al.*, 1994] W. Neuenschwander, P. Fua, G. Székely, and O. Kubler. Making Snakes Converge from Minimal Initialization. In *Proceedings of the DARPA Image Understanding Workshop*, Monterey, CA, November 1994. Morgan Kaufmann.
- [Quam, 1978] L.H. Quam. Road Tracking and Anomaly Detection. In *Proceedings of the DARPA Image Understanding Workshop*, pages 51–55. Morgan Kaufmann, May 1978.