The UMass Ascender System for 3D Site Model Construction*

Robert T. Collins, Christopher O. Jaynes, Yong-Qing Cheng, XiaoGuang Wang,
Frank R. Stolle, Howard Schultz, Allen R. Hanson, Edward M. Riseman
Computer Vision Laboratory
Computer Science Department
University of Massachusetts, Amherst
Amherst, Massachusetts 01003

Abstract

The Automated Site Construction, Extension, Detection and Refinement system (Ascender) has been developed to automatically populate a site model with 3D buildings extracted from multiple, overlapping views. Image Understanding (IU) algorithms hypothesize potential building roofs in one image, automatically locate supporting geometric evidence in other images, and determine the precise shape and position of the new buildings via multi-image triangulation. Backprojecting image intensities onto the recovered object surfaces leads to realistic graphical site model displays, and to extraction of symbolic features such as windows and doors. This paper describes how the Ascender system acquires and extends an initial site model. System performance is evaluated using imagery from Fort Hood, Texas.

1.0 Introduction

The University of Massachusetts has a long-term commitment to the development of knowledge-based computer vision systems, and the UMass RADIUS project is the latest example of this focus. Image understanding modules have been developed to acquire, extend and refine 3D volumetric building models from multiple, overlapping aerial views. The system design emphasizes model-directed processing, rigorous camera geometry, and fusion of information across multiple images for increased accuracy and reliability.

Site *model acquisition* involves processing a set of images to detect man-made and natural features of interest, and to determine their 3D shape and placement in the scene. The site models produced have obvious applications in areas such as surveying, surveillance and automated cartography. For example, acquired site models can be used for automated model-to-image registration of new images [Collins93], allowing features in the model to be overlaid on the images to aid visual change detection. Two other important site modeling tasks are *model extension*, updating the geometric site model by adding or removing features (Section 4), and *model refinement*, iteratively refining the shape and placement of features as more views become available. Model extension and refinement are ongoing processes applied whenever new images become available, each updated model becoming the current site model for the next iteration. Thus, over time, the site model is steadily improved to become more complete and more accurate.

The Ascender system has been developed to automatically extract buildings from multiple, overlapping images of a site. Since buildings come in all sizes and shapes, an initial generic class of flat-roofed, rectilinear buildings was chosen to maintain tractable implementation goals. This class contains all buildings where pairs of adjacent roof edges are perpendicular and lie in a single, horizontal plane; the simplest examples are rectangular box-shapes, L-shapes, and U-shapes. The most prevalent building types not included in this class are peaked-roof and multi-level structures.

2.0 Automatic Model Acquisition

Acquisition of an initial site model begins with a set of overlapping images for which intrinsic and extrinsic camera parameters are known. We break the processing into five distinct stages that each

*Funded by the RADIUS project under ARPA/Army TEC contract number DACA76-92-C-0041, and the National Science Foundation grant number CDA8922572.

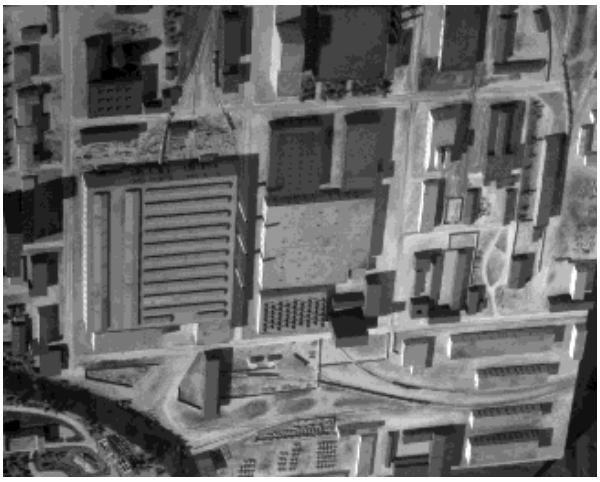


Figure 1. A sample image from Model Board 1

contribute to producing the final, three-dimensional model:

1) line segment extraction,
2) rooftop polygon detection,
3) multi-image epipolar matching,
4) constrained, multi-image triangulation, and
5) projective intensity mapping.

Brief descriptions of each of these component tasks are presented; detailed algorithmic descriptions are outside the scope of this paper, and references to additional material are provided for the interested reader. The algorithms are illustrated using a running example performed using images J1-J8 from the RADIUS Model Board 1 data set. Figure 1 shows a sample image from the data set. The scene is a 1:500 inch scale model of an industrial site, with ground truth measurements available for about 110 points scattered throughout the model. The scale model is built on a table top that can be raised and tilted to simulate a variety of camera altitudes and orientations. For model board images J1-J8 the table was set to simulate aerial photographs taken with a ground sample distance of 18 inches, that is, pixels near the center of the image backproject to quadrilaterals on the ground with sides approximately 18 inches long (all measurements will be reported in scaled-up object space coordinates). Each image contains approximately 1320x1035 pixels, with roughly 11 bits of grey level information per pixel. The dimensions of each image vary slightly because the images have been subjected to unmodeled geometric and photometric distortions that were intended to simulate actual operating conditions.

2.1 Line Segment Extraction

To help bridge the huge representational gap between pixels and site models, feature extraction routines are applied to produce symbolic, geometric representations of potentially important image features. The Ascender system relies on straight line segments extracted by the Boldt algorithm [Boldt89], developed at UMass. At the heart of the Boldt algorithm is a hierarchical grouping system inspired by the Gestalt laws of perceptual organization. Zero-crossings of the Laplacian of the intensity image provide an initial set of local intensity edges. Hierarchical grouping then proceeds iteratively; at each iteration edge pairs are linked and replaced by a single longer edge if their end points are close and their orientation and contrast values are similar. The resulting line set can be filtered according to user-supplied length and contrast thresholds that can be set interactively by the IA using one-dimensional slider bars.

The Boldt line algorithm is employed in this project because of its precision and sensitivity, even though it dominates the computation time for the entire model acquisition process. The Boldt line detector, run on a typical modelboard subimage of 512x512 pixels takes approximately 14 minutes of CPU time on a Sun Sparc-20 workstation. Since significant off-line computation is allowable for initial model acquisition, and since the accuracy of the derived three-dimensional models depends directly on the two-dimensional accuracy of extracted line features, we continue to use the Boldt algorithm even though faster (but less accurate) line extraction algorithms are available. The current implementation of the Boldt algorithm cannot directly handle the memory requirements of the very large images that are common in the RADIUS application domain (even the 1320x1035 model board images are fairly small in comparison). Therefore, large images are cut into overlapping subimages, each of which is processed separately to extract line segments. All lines found are then translated and scaled back into the original image coordinate system. Breaking the image into overlapping pieces introduces some artifacts into the line data. In particular, lines are fragmented at subimage boundaries, and lines lying totally within an overlapping area are duplicated. However, all of the building extraction algorithms that use line segments are built to operate under the assumption of

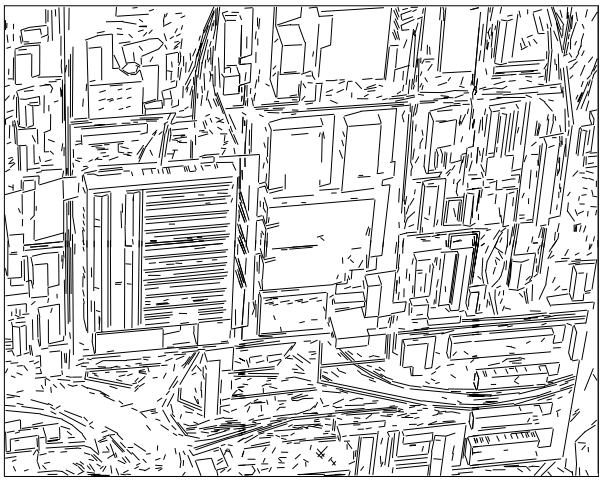


Figure 2. Line segments extracted from Figure 1

noisy, fragmented data. Thus, although no attempt has been made to post-process the line data to remove these artifacts, the performance of the building detection process does not appear to be degraded.

For model board images J1-J8, subimages of size 266x266 with an overlap of 10 pixels were used. For the results presented here, image resolution was also reduced by half using Gaussian filtering and subsampling, even though full resolution is used by default in the Ascender system (in this example, it was found that Gaussian image reduction eliminated the peculiar "sawtooth" noise pattern that corrupted some of the model board images). The line segments were filtered to keep lines longer than 10 pixels that have a contrast of at least 15 gray levels. This resulted in roughly 2800 line segments per image. Figure 2 shows a representative set of line segments extracted from the image shown in Figure 1.

2.2 Building Rooftop Detection

The goal of automated rooftop detection is to roughly delineate building boundaries that will later be verified in other images and triangulated to create 3D geometric building models. The rooftop detection algorithm is based on finding image polygons corresponding to the boundaries of flat, rectilinear rooftops in the scene (for more details, see [Jaynes94]). Briefly, possible roof corners are identified by pairs of line segments with spatially proximate endpoints, meeting at an angle that could correspond to the projection of a horizontal, orthogonal roof corner in the scene. Perceptually compatible corner pairs are linked with surrounding line data and entered into a feature-relation graph (Figure 3), weighted according to the amount of support they receive from the low-level image data. Potential building roof polygons appear as cycles in the graph, and virtual corner features may be hypothesized automatically to complete a cycle, if necessary. Rooftops are finally extracted by partitioning the feature-relation graph into a set of maximally weighted, independent cycles representing closed, high-confidence building roofs.

The building detector was run on image J3 in this example. This is a near-nadir view, but nothing in the code precludes using one of the oblique views instead. Since rooftop detection is computationally expensive due to low-level feature extraction and the rapid growth of the feature-relation graph with image size, the image was partitioned manually into nine separate regions, loosely representing different "functional areas". The roof detector generated 40 polygonal rooftop hypotheses, shown in Figure 4. Most of the hypothesized roofs are rectangular, but six are L-shaped. Note that the overall performance is quite good for buildings entirely in view. Most of the major roof boundaries in the scene have been extracted, and in the central cluster of buildings (see area **A** in Figure 4) the segmentation is nearly perfect.

There were some false positives - polygons extracted that do not in fact delineate the boundaries of a roof. The most obvious example is the set of overlapping polygonal rooftops detected over the large building with many parallel roof vents (marked **B** in Figure 4). Nevertheless it should be noted that the correct outer boundary of this building roof is detected. The set of parallel

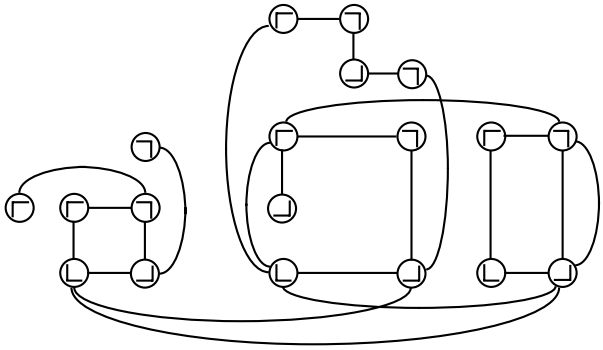


Figure 3. Corners and lines are represented in a feature relation graph with perceptually compatible corners linked.

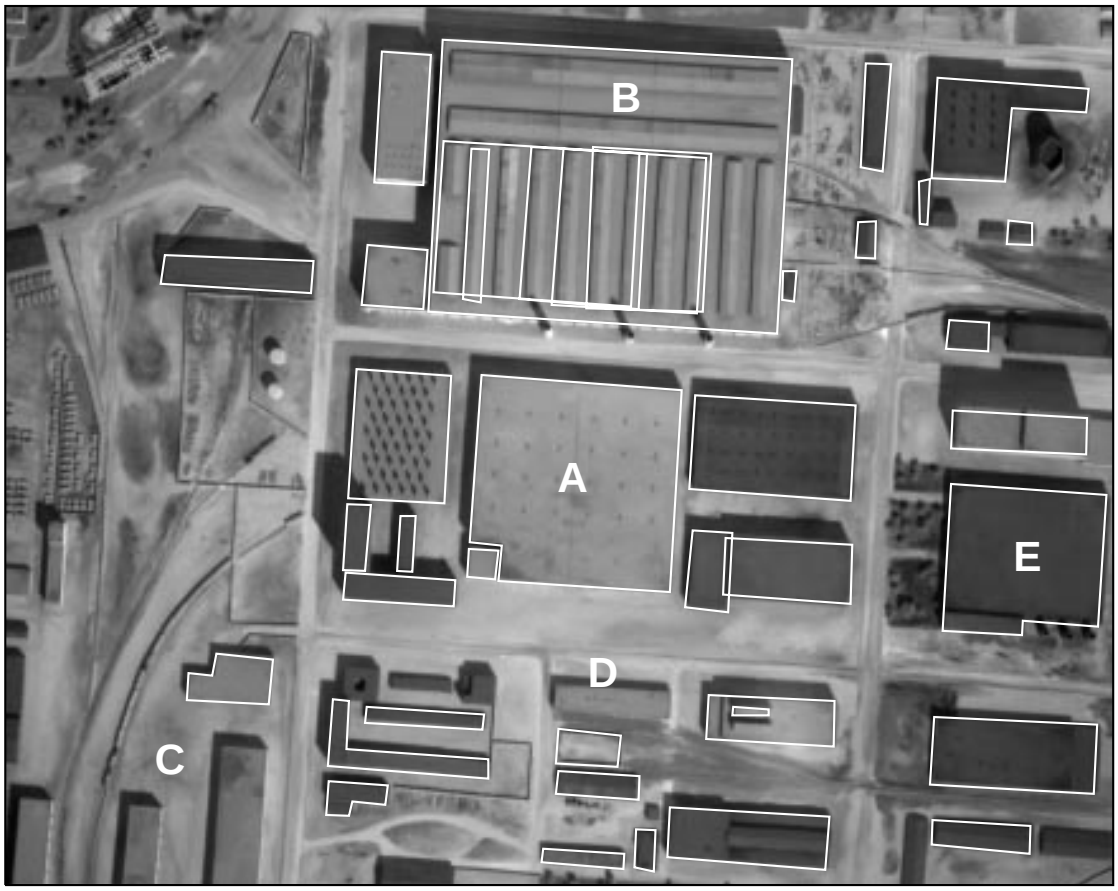


Figure 4. Rooftop hypotheses extracted from model board image J3. Alphabetic labels are referred to in the text.

roof vents on this building, coupled with the close proximity of other buildings and three tall smokestacks (and their shadows!) that occlude and fragment the building boundary in many of the images, make this one of the most challenging buildings in the site for rooftop detection, epipolar matching, and intensity mapping. The detection of these roof vents typifies a common problem of false alarms, where significant rooftop details are extracted as rooftop hypotheses in their own right. At the level of 2D polygon extraction, it is difficult to provide an effective method to distinguish between roofs and roof substructures; however, three-dimensional information introduced in subsequent stages of processing may provide the additional information necessary for making this decision.

There are also some false negatives, which are buildings that should have been detected, but were not. The most prevalent example of this is a set of buildings (see **C**) that are only partially in view at the edge of the image. Since image boundaries are not used as possible lines, these buildings cannot possibly generate complete polygons, and thus should not really be considered as failures. Although the current system is designed to detect only complete building models, subsequent epipolar feature matching and multi-image line triangulation routines are able to handle partial building "fragments", and therefore future control strategies could be developed to allow merging of partial building wireframes produced from different images into a single building model [Jaynes, companion chapter in this book]. Label **D** marks a false negative that is in full view. Two adjacent corners in the rooftop polygon were missed by the corner extraction algorithm. Although a top-down virtual feature hypothesis can be invoked to insert a single missing corner in an incomplete rooftop polygon, there is currently no recovery mechanism when two adjacent corners are missing. It should be stressed that even though a single image was used here for bottom-up hypotheses, buildings that are not extracted in one image will often be found easily in other images with different viewpoints and sun angles (see Section 4).

There are several cases that cannot be strictly classified as false positives or false negatives. Several split-level buildings appearing along the right edge of the image (for example **E** and the building above) are outlined with single polygons rather than with one polygon per roof level. Some peaked roof buildings were also outlined (at the bottom right, for example), even though they do not conform to the assumptions underlying this version of the system.

2.3 Multi-image Epipolar Matching

After detecting a potential rooftop in one image, corroborating geometric evidence is sought in other images (often taken from widely different viewpoints) via epipolar feature matching. Rooftop polygons are matched by searching for each component 2D line segment separately and then fusing the results. For each polygon segment from one image, an epipolar search area is formed in each of the other images, based on the known camera projection equations and the assumption that the roof is flat. This quadrilateral search area is scanned for possible matching line segments, each potential match implying a different height of a 3D roofline in the scene (see Figure 5). Results from each line search are combined in a one-dimensional histogram, with each 2D match voting for a range of 3D roof heights centered at each match, weighted by compatibility of the match in terms of expected line segment orientation and length. This approach allows fragmented line data to be handled correctly without any knowledge or additional processing, since the combined votes of all subpieces of a fragmented line count the same as the vote of a full-sized, unfragmented line.

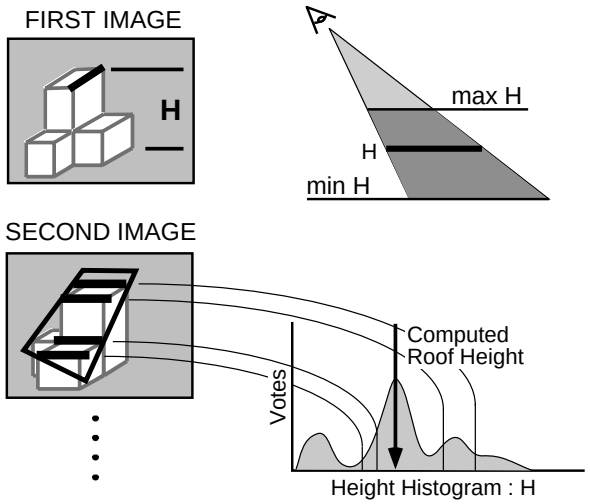


Figure 5. Matches of a reference line segment in one image are found in other images. Each match contributes a Guassian-weighted vote into a 1D height histogram for the roof surface.

A single global histogram accumulates height votes from multiple edges in a rooftop polygon across multiple images. After all votes have been tallied, the histogram bucket containing the most votes yields an estimate of the 3D height of the roof polygon in the scene, as well as the set of correspondences between rooftop edges and image line segments in multiple views. The reader should note that building rooftops are assumed to be horizontal relative to the ground plane, thus the matching algorithm is able to use a simple one-dimensional height histogram for collecting votes. The approach is not limited to this assumption, however, and can be extended to vote for planar surfaces at other orientations using a multidimensional histogram.

Minimum and maximum values for the epipolar height histogram are chosen by the user based on whatever collateral or assumed knowledge is available. These set the bounds for the epipolar search region, and if they are set too wide could reduce the likelihood of finding a clear, unique peak due to the introduction of numerous incorrect matches. For the Model Board 1 experiment, the minimum and maximum height values were set at -92 ft and 146 ft, respectively, and the histogram contained 24 buckets with a height range of roughly 12 feet per bucket. After epipolar voting is completed for a rooftop polygon, line segment correspondences are extracted from the histogram bucket containing the highest number of votes and those buckets immediately adjacent to it. Epipolar matching of a rooftop hypothesis is considered to have failed when, for any edge in the rooftop polygon, no line segment correspondences are found in any image. Based on this criterion, epipolar matching failed on eight rooftop polygons in this example. Six were either peaked or multi-layer roofs that did not satisfy our assumption of generic flat-roofed building models, and the other two were building fragments with some sides shorter than the minimum length threshold applied to the line segment data. When multi-level or peaked roof buildings vote in the epipolar matching process, their votes get diluted due to lines occuring at different heights, and therefore no clear peak can be found. This automati-

cally filters out buildings that do not have flat roofs. Note, however, that epipolar matching can also be applied to individual polygon edges rather than all of them, leading to a mechanism for extracting horizontal lines at different heights along the boundaries of multi-level roofs [Jaynes, attached companion paper].

At this stage in the experiment we also removed six obviously incorrect building hypotheses by hand. Five of them comprised the set of overlapping polygons within the building labeled **B** in Figure 4. The sixth was the fenced in area appearing directly below label **D** in that image. Typically, false hypotheses intersect correct hypotheses and a simple arbitration scheme can be used to remove the conflicting building model with the lowest confidence measure. Detecting and removing such conflicts automatically in this manner is a recent improvement to the system that was not in place at the time this example was run.

2.4 Multi-image Line Triangulation

Multi-image triangulation is performed to determine the precise size, shape, and position of a building in the local 3D site coordinate system. Object-level constraints such as perpendicularity and coplanarity are imposed on the solution to assure reliable results. This algorithm is used for triangulating 3D rooftop polygons from the line segment correspondences determined by epipolar feature matching.

The parameters estimated for each rooftop polygon are shown in Figure 6. The horizontal plane containing the polygon is parameterized by a single variable ***Z***. The orientation of the rectilinear structure within that plane is represented by a single parameter θ. Finally, each separate line within the polygon is represented by a single value $\mathbf{r_i}$ representing the signed perpendicular distance of that line from some nominal point in the plane. The representation is simple and compact, and the necessary coplanarity and rectangularity constraints on the polygon's shape are built in. (A more general approach based on the Plucker coordinate representation of 3D lines has also been implemented for triangulating general wireframe structures [Cheng94]).

A standard Levenberg-Marquardt algorithm is employed to determine the set of polygon parameters that minimize an objective "fit" function that measures how well each projected rooftop edge aligns with the 2D image segments that correspond to it. Such nonlinear estimation algorithms typically require an initial estimate that is then iteratively refined. In this system, the original 2D rooftop polygon extracted by the building detector, and the roof height estimate computed by the epipolar matching algorithm, are used to generate the initial flat-roofed polygon estimate. After triangulation, each refined 3D roof polygon is then extruded down to the ground to form a volumetric model. For the Model Board 1 site, the ground was represented as a horizontal plane with Z-coordinate value determined from the ground truth measurements. More generally, the system can use digital elevation maps produced by the UMass Terrain Reconstruction System [Schultz94], or any other available terrain map that is sufficiently accurate.

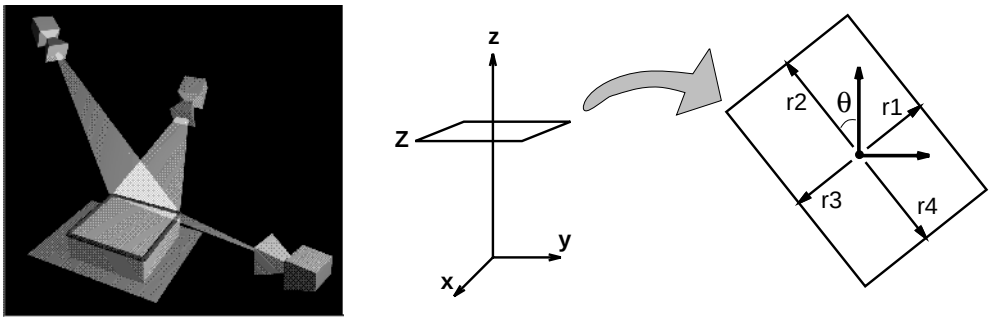


Figure 6. (a) Multiple views are used as input to constrain a nonlinear estimation algorithm that enforces geometric constraints of perpend-icularity and coplanarity. (b) Parameterization of a flat, rectlinear polygon for multi-image rooftop triangulation.

Outlines of the final set of triangulated rooftops are shown in Figure 7. The rightmost polygon in the image is noticeably incorrect. This polygon actually corresponds to a split-level building containing two roofs at different heights in the scene. Most multi-level buildings were automatically filtered out during epipolar matching, but this one managed to survive. To evaluate the 3D accuracy of the triangulated building polygons, 21 roof vertices were identified where ground truth measurements are known (numbered vertices in Figure 7). The average Euclidean distance between triangulated polygon vertices and their ground truth locations is 4.31 feet, which is reasonable given the level of artificially introduced geometric distortion present in the images. The average horizontal distance error is 3.76 feet, while the average vertical error is only 1.61 feet. This is understandable, since all observed rooftop lines are considered simultaneously when estimating the building height (vertical position), whereas the horizontal position of a rooftop vertex is primarily affected only by its two adjacent edges. In most cases, the vertical position of a vertex is constrained by at least twice as many lines as the horizontal position.

2.5 Projective Intensity Mapping

Backprojection of image intensities onto polygonal faces of building models enhances their visual realism and also provides a convenient storage mechanism for later symbolic extraction of detailed surface structure. Planar projective transformations provide a mathematical description of how surface structure from a planar building facet maps into a perspective image. By inverting this transformation using known building position and camera transformations, intensity information from each image is backprojected to "paint" the walls and roof of the building model. Since multiple images are used, intensity information from many (sometime all) faces is available, even though they are not all visible from any single view. The resulting intensity-mapped site model can then be rendered to predict how the scene will appear from a new view, and on high-end workstations realistic real-time "fly-throughs" are achievable.

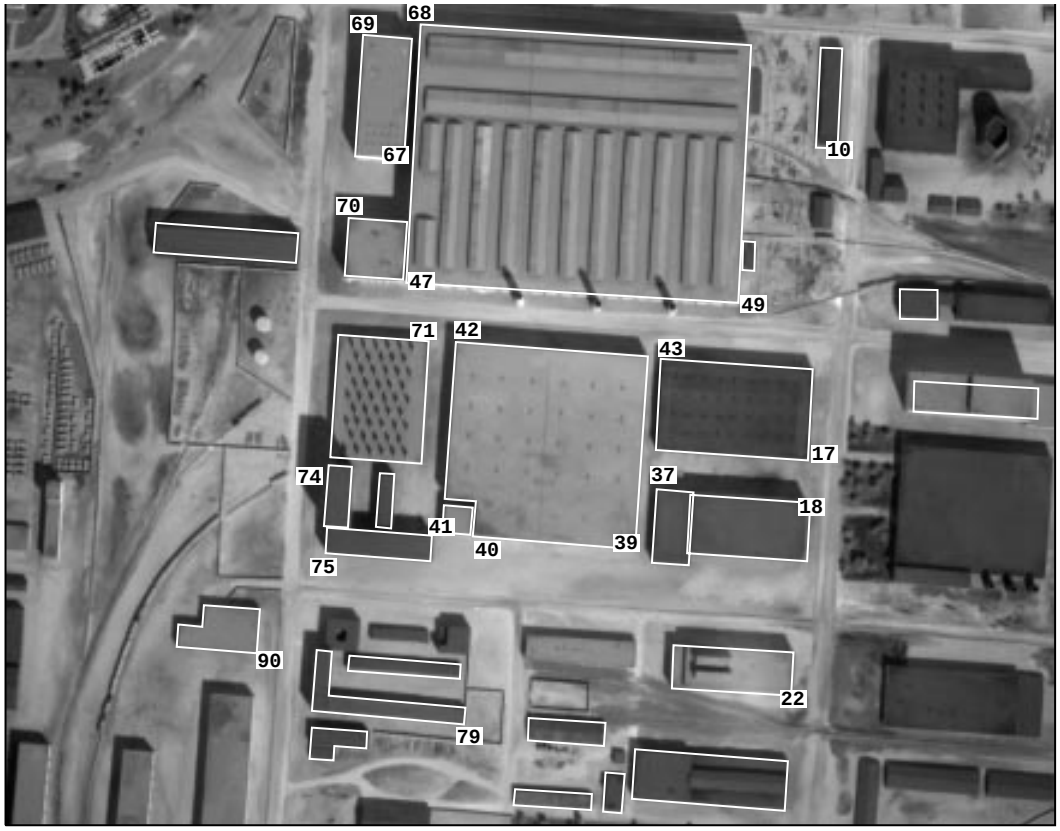


Figure 7. Reprojection of 3D triangulated rooftops back into image J3. Numerical labels mark roof vertices where ground truth measurements are known.

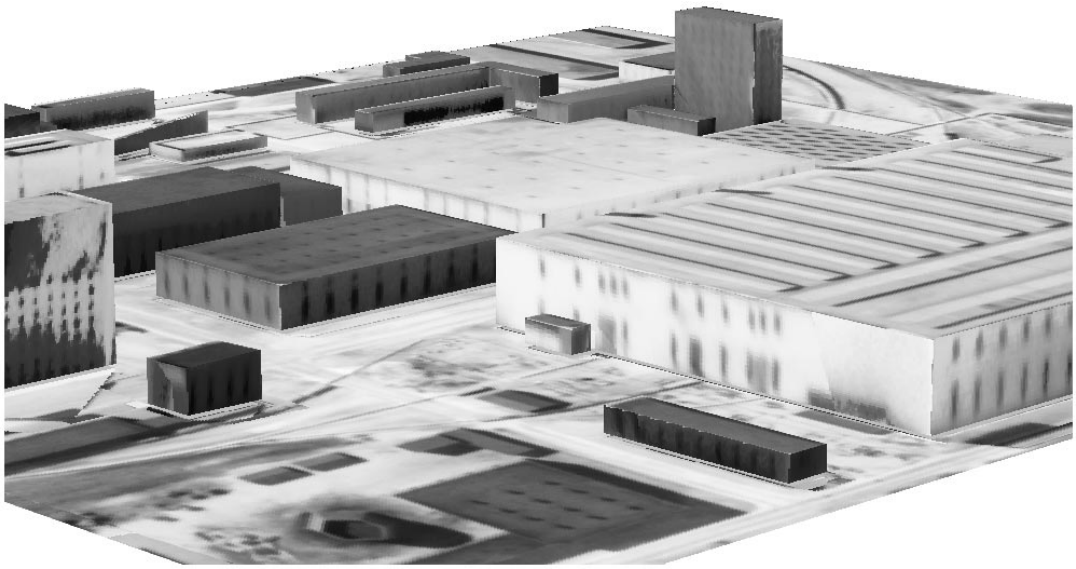


Figure 8. Rendered view of the acquired Model Board 1 site model.

For each of the 25 volumetric building models extracted from Model Board 1, a set of intensity maps was generated for each planar facet by projectively mapping intensity values from the images in which the facet is visible. When multiple images showing a single building facet are available (the typical case for building rooftops), a composite intensity map is automatically generated to synthesize the best available image information at each point in terms of resolution and contrast. This process is described in detail in [Wang96]. Figure 8 shows an example of a generated site display of Model Board 1 using automatically derived building models and intensity maps.

Although intensity mapping enhances the virtual realism of graphic displays, this illusion of realism is greatly reduced as the observer's viewpoint comes closer to the rendered object surface, or as the viewing orientation becomes significantly different than the sensor viewpoint used to produce the texture map. For example, a highly oblique view of a wall will not produce effective texture maps for generating perpendicular views. What is needed to go beyond simple intensity mapping is explicit, symbolic extraction and graphical model insertion of detailed surface structures such as windows, doors and roof vents. Backprojected intensity maps provide a convenient starting point, since rectangular lattices of windows or roof vents can be searched for without complication from the effects of perspective distortion, and model-based extraction of surface structure can be applied only where relevant, i.e. window and door extraction can be focused on wall intensity maps, while roof vent computations are performed only on roofs. As one example, a generic algorithm has been developed for extracting windows and doors on wall surfaces, based on a rectangular region growing method applied at local intensity minima in the unwarped intensity map. Extracted window and door hypotheses are used to compose a refined building model that explicitly represents those architectural details. An example is shown in Figure 9. The windows and doors have been rendered as dark and opaque, but since they are now symbolically represented, it would be possible to render the windows with glass-like properties such as transparency and reflectivity.

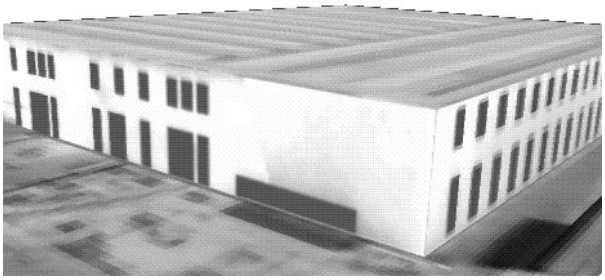


Figure 9. Detailed building model rendered using symbolically extracted windows and doors.

3.0 Evaluation on Fort Hood

The success of the Ascender system will ultimately be judged by its performance on classified imagery. As of this writing, a series of such tests is being performed at Lockheed-Martin and at the National Exploitation Lab (NEL). In parallel with that effort, UMass is performing an in-depth, quantitative system evaluation using unclassified data. This section summarizes the results of an evaluation on a large data set from Fort Hood, Texas. A more detailed description of this experiment can be found in [Collins96].

An evaluation data set was cropped from the Fort Hood imagery to yield seven subimages with varying viewpoints (two nadir views: 711 and 713, two slightly off-nadir views: 525 and 927, and three oblique views: 1025, 1125 and 1325). Ground sample distances are roughly 0.3 meters for the nadir views, 0.6 meters for the off-nadir views, and 1.0 meter for the oblique views. The region of overlap for the evaluation area covers roughly 760x740 meters, containing a good blend of both simple and complex roof structures. Thirty ground truth building models were created by hand using interactive modelling tools provided by the RADIUS Common Development Environment (RCDE) [Mundy92]. Each building is composed of RCDE "cube", "house" and/or "extrusion" objects that were shaped and positioned to project as well as possible (as determined by eye) simultaneously into the set of seven images. The ground truth data set is overlaid on one of the nadir views in Figure 10. Some 3D building models produced automatically by the Ascender system on this dataset are shown in Figure 11.

Since the Ascender system explicitly recovers only rooftop polygons (the rest of the building wireframe is formed by vertical extrusion), the evaluation is based on comparing detected 2D and triangulated 3D roof polygons vs. their ground truth counterparts. There are 73 ground truth rooftop polygons among the set of 30 buildings. Ground truth 2D polygons for each image are determined by projecting the ground truth 3D polygons into that image using the known camera projection equations.

3.1 Evaluation of 2D Detection Rates

One important module of the Ascender system is the 2D polygonal rooftop detector. The detector

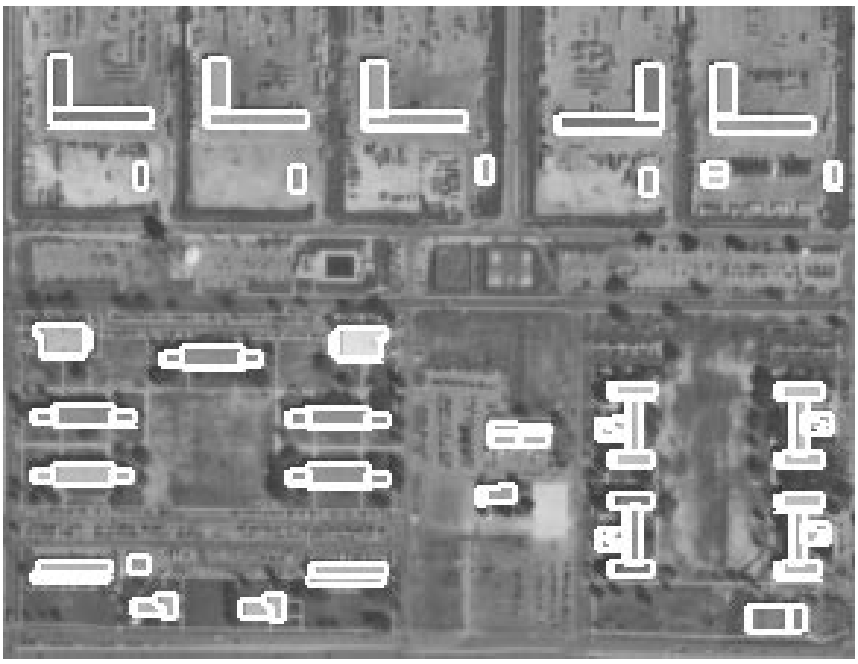


Figure 10. Fort Hood evaluation area with 30 ground truth building models. There are 73 roof facets in all.

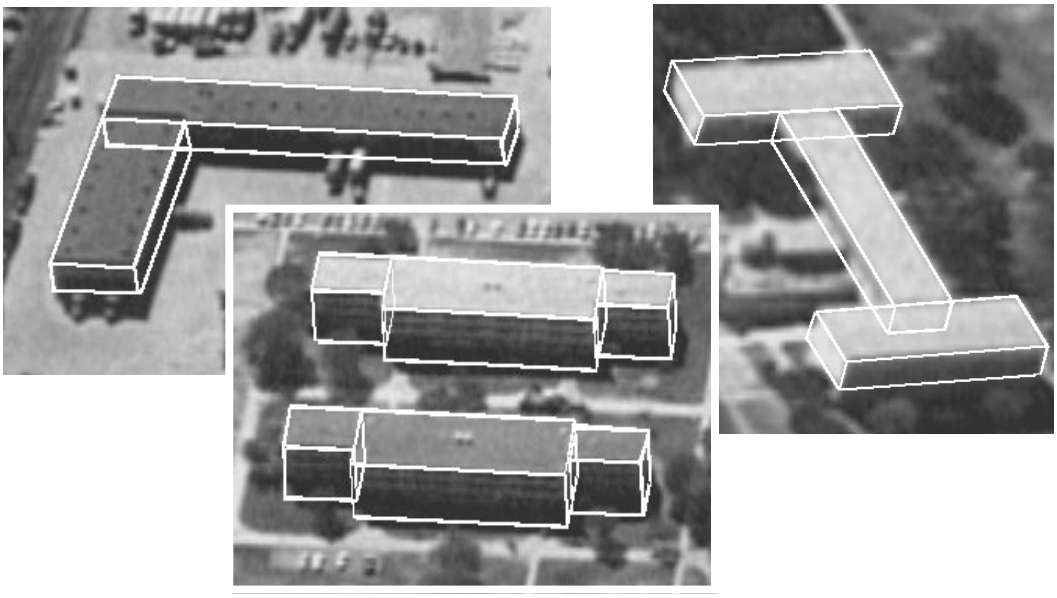


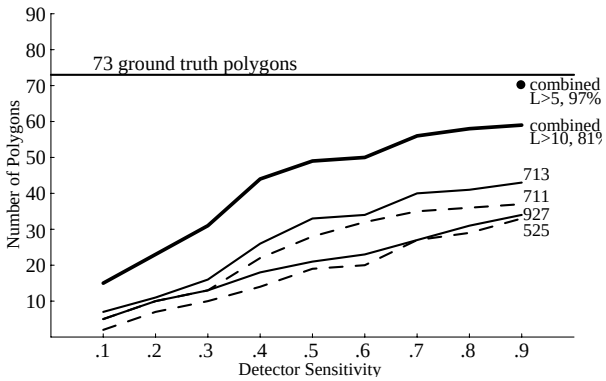
Figure 11. Sample building models extracted automatically by the Ascender system.

was tested on four images from the test set (two nadir, two slightly off-nadir) to see how well it performed at different grouping sensitivity settings, and with different length and contrast settings of the Boldt line extraction algorithm. The detector was tested by projecting each ground truth roof polygon into an image, growing its 2D bounding box out by 20 pixels on each side, then invoking the building detector in that region to hypothesize 2D rooftop polygons. The evaluation goal of this experiment was to determine the true positive detection rate *when the building detector was invoked on an area containing a building*.

Graph 1 plots the number of true positive hypotheses produced by the building detector on each of the four images, for nine different sensitivity settings ranging from 0.1 to 0.9 (very low to very high) that control the perceptual grouping process. The line segments used were computed by the Boldt algorithm with length and contrast thresholds of 10. This set of lines was deemed a reasonable trade-off between reducing computation in generating the initial feature set of lines and extracting the required information for building detection. For the highest sensitivity setting, the percentage of rooftops detected was 51% and 59% respectively for the two nadir views, and 45% and 47% for the two off-nadir views. The graph also shows the number of true positives achieved by combining the hypotheses from all four images, either by pooling hypotheses computed separately for each image, or by recursively masking out previously detected buildings and focusing on the unmodeled areas in each new image. For the highest sensitivity setting, this strategy detects 81% (59 out of 73) of the rooftops in the scene.

To measure the best possible performance of the rooftop detector on this data, it was run on all four images at sensitivity level 0.9, using a more complete set of Boldt line data computed with length and contrast thresholds of 5. These were judged to be the highest sensitivity levels for both line extractor and building detector that were feasible, and the results represent the best possible perfor-

Graph 1. Building detector sensitivity vs. number of true positives. The horizontal line marks the actual number of ground truth polygons. Combining results from all four views yields a "best" detection rate of 81% for Boldt lines of length > 10, and 97% with lines of length > 5.



mance of the building detector on each image. The percentages of rooftops detected in each of the four images under these conditions were 86%, 84%, 74%, and 67%, with a combined image detection rate of 97% (71 out of 73).

3.2 Evaluation of 3D Accuracy

The second major subsystem in Ascender takes 2D roof hypotheses detected in one image and reconstructs 3D rooftop polygons via multi-image line segment matching and triangulation. The final reconstruction accuracy depends on the number and geometry of the views used, and also on the 2D image accuracy of the hypothesized roof polygons. An evaluation experiment was performed to determine the typical end-to-end performance of the system by taking the true positive 2D polygons detected in the last section for each of the four views, and performing matching and triangulation using the other six views.

Distances between 3D vertices of each reconstructed 3D building roof and the corresponding ground truth roof vertices were measured, and decomposed into their planimetric (horizontal) vs. altimetric (vertical) components. The median distance errors are shown in Table 1, broken down by the image in which the 2D polygons feeding the reconstruction process were hypothesized. The results suggest that the planimetric component of reconstructed vertices is more sensitive to inaccuracies in the detection and triangulation process than the altimetric component. This result is consistent with previous observations that the corners of Ascender's reconstructed building models are more accurate in height than in horizontal position (Section 2.4).

Table 1. Median planimetric and altimetric errors (in meters) between reconstructed 3D polygon vertices and ground truth roof vertices.

	711	**713**	**525**	**927**
planimetric	0.68	0.73	1.09	0.89
altimetric	0.51	0.55	0.90	0.61

3.3 Evaluation Summary

Further experimental results from the Fort Hood evaluation can be found in [Collins96]. Based on the results of the evaluations to date, we can make the following quantitative statements about how the system is expected to perform under different scenarios.

Single-Image Performance: The building detection rate varies roughly linearly with the sensitivity setting of the polygon detector. At the highest sensitivity level, roughly 50% of the buildings are detected in each image using Boldt lines extracted at a medium level of sensitivity (length and contrast > 10), and about 75-80% when using Boldt lines extracted at a high level of sensitivity (length and contrast > 5). The increased sensitivity produces significantly more lines with a commensurate increase in computation, but is definitely worthwhile for the RADIUS application domain). Nadir views appear to produce better detection rates than obliques, but this can be explained by large differences in GSD for this image set and may not be characteristic of system performance in general.

Multiple-Image Performance: One of our underlying research hypotheses is that the use of multiple images increases the accuracy and reliability of the building extraction process. Rooftops that are missed in one image are often found in another, so combining results from multiple images typically increases the building detection rate. By combining detected polygons from four images, the total building detection rate increased to 81% using medium-sensitivity Boldt lines, and to 97% using high-sensitivity ones. For this data set, 3D building corner positions were recovered with an accuracy of better than one meter. We have observed in further experiments that matching and triangulation to produce 3D roof polygons, and thus the full building wireframe by extrusion, can perform at satisfactory levels of accuracy given only a pair of images, but using three views gives noticeably better results (roughly a 20% increase in 3D accuracy over using two views). After four images, only a modest increase in 3D accuracy is gained (using four images rather than three yields an additional 10% increase in accuracy, but adding a fifth image yields only a 4% increase over that).

4.0 Site Model Extension

After an initial site model has been acquired, it may be necessary to periodically update it based on new imagery. New buildings may have been built at the site, and old buildings may have been destroyed. This section briefly addresses the problem of extending a site model database to include buildings that were previously unmodeled, either because they were not detected in previous images, or because they were recently constructed. The main difference between model extension and model acquisition is that now a partial site model exists that can be used to guide the processing of new images. In particular, camera pose for each image can be determined via model-to-image registration. Our approach to model-to-image registration involves two components: model matching and pose determination.

The goal of *model matching* is to find the correspondence between 3D features in a site model and 2D features that have been extracted from an image; in this case determining correspondences between edges in a 3D building wireframe and 2D extracted line segments from the image. The model matching algorithm described in [Beveridge95] is used. Based on a local search approach to combinatorial optimization, this algorithm searches the discrete space of correspondence mappings between model and image line features for one that minimizes a match error function. The match error depends upon how well the projected model geometrically aligns with the data, as well as how much of the model is accounted for by the data. The result of model matching is a set of correspondences between model edges and image line segments, and an estimate of the transformation that brings the projected model into the best possible geometric alignment with the underlying image data.

The second aspect of model-to-image registration is precise *pose determination*. It is important to note that since model-to-image correspondences are being found automatically, the pose determination routine must take into account the possibility of outliers (gross mistakes) in the set of correspondences found. The robust pose estimation procedure described in [Kumar94] is used. At the heart of this code is an iterative, weighted least-squares algorithm for computing pose from a set of correspondences that are assumed to be free from outliers. The pose parameters are found by minimizing an objective function that measures how closely the projected model features map to their corresponding image features. Since it is well known that least squares optimization techniques can fail catastrophically when outliers are present in the data, this basic pose algorithm is embedded inside a least-median-squares (LMS) procedure that repeatedly samples subsets of correspondences to find one devoid of outliers. LMS is robust over data sets containing up to 50% outliers. The final results of pose determination are a set of camera pose parameters and a covariance matrix that estimates the accuracy of the solution.

This process is illustrated using the partial site model constructed in Section 2, and image J8 from the Radius Model Board 1 dataset. Results of model-to-image registration of image J8 with the partial site model can be seen in Figure 12, which shows projected building rooftops from the site model (thin lines) overlaid on the image. It is immediately apparent that the current model does not account for all the building structures in the new image. To extend the site model, image areas containing known buildings were masked off, and the Ascender system was run on the unmodeled areas to hypothesize an additional set of 3D volumetric building models. These were added to the site model database to produce the extended model, shown projected into Figure 12 (thick lines). Most of the buildings that remain unmodeled are located at the periphery of the site. Since this area is not visible in many of the eight views, these buildings fail to generate the necessary multi-image evidence that is needed for construction of a high-confidence 3D building hypothesis. If more images were used with greater site coverage, most of these buildings would appear in the 3D site model.

5.0 Conclusion

An extensive research effort is underway at UMass to develop capabilities for automated 3D site modeling from aerial images. The Ascender system has been developed to extract and model flat-roofed, rectilinear buildings from multiple views. Version 1.0 of Ascender has been delivered to Lockheed-Martin for testing on classified imagery and for

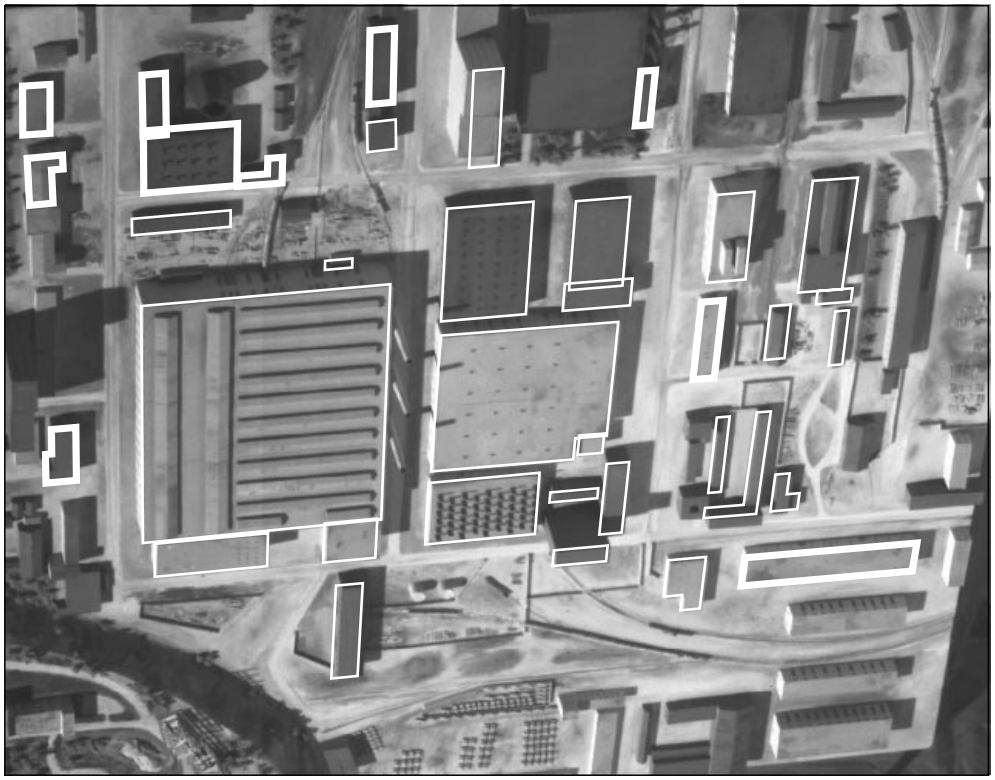


Figure 12. Updated site model projected onto image J8. Thin lines denote the registered partial site model. Thick lines delineate buildings that were automatically detected and added via model extension.

integration into the RADIUS Testbed. An evaluation of Ascender on an unclassified data set of Fort Hood has been performed at UMass. The results suggest that the system performs reasonably well in terms of detection rate and accuracy, and that performance degrades gracefully when the number of images used is small. Much more testing will be needed to determine how the system performs under various weather and viewing conditions, in order to formulate a set of recommendations as to how and when to use the system.

Algorithms and strategies for extracting other common building classes with peaked, curved and multi-level flat roofs are currently being developed and tested for eventual inclusion into future versions of Ascender. Much of this new research is described in [Jaynes, companion paper in this volume]. Moving beyond a single control strategy for detecting a single class of buildings brings to the forefront complex issues of context-sensitive control strategies, model class selection, data fusion, and hypothesis arbitration, and these topics are the focus of our current research efforts.

Bibliography

[Beveridge95] J.R.Beveridge and E.Riseman, "Optimal Geometric Model Matching under Full 3D Perspective," *CVGIP: Image Understanding*, Vol.61(3), 1995, pp.351-364.

[Boldt89] M.Boldt, R.Weiss and E.Riseman, "Token-Based Extraction of Straight Lines," *IEEE Transactions on Systems, Man and Cybernetics*, Vol.19(6), 1989, pp.1581-1594.

[Cheng94] Y.Cheng, R. Collins, A. Hanson and E. Riseman, "Triangulation Without Correspondences," *ARPA Image Understanding Workshop*, Monterey, CA, 1994, pp.993-1000.

[Collins96] R.Collins, A.Hanson, E.Riseman, C.Jaynes, F.Stolle, X.Wang, and Y.Cheng, "UMass Progress in 3D Building Model Acquisition," *ARPA Image Understanding Workshop*, Palm Springs, CA, Feb.1996, pp.305-315.

[Collins93] R.Collins, A.Hanson, E.Riseman and Y.Cheng, "Model Matching and Extension for Automated 3D Site Modeling," *DARPA Image*

Understanding Workshop, Washington, DC, April 1993, pp.97-203.

[Jaynes94] C.Jaynes, F.Stolle and R.Collins, "Task Driven Perceptual Organization for Extraction of Rooftop Polygons," *IEEE Workshop on Applications of Computer Vision*, Sarasota, FL, Dec.1994, pp.152-159.

[Kumar94] R.Kumar and A.Hanson, "Robust Methods for Estimating Pose and Sensitivity Analysis," *CVGIP: Image Understanding*, Vol.60(3), Nov.1994, pp.313-342.

[Mundy92] J.Mundy, R.Welty, L.Quam, T.Strat, W.Bremner, M.Horwedel, D.Hackett and A.Hoogs, "The RADIUS Common Development Environment," *DARPA Image Understanding Workshop*, San Diego, CA, Jan.1992, pp.215-226.

[Schultz94] H.Schultz, "Terrain Reconstruction from Oblique Views," *ARPA Image Understanding Workshop*, Monterey, CA, Nov.1994, pp.1001-1008.

[Wang96] X.Wang, W.J.Lim, R.Collins and A.Hanson, "Automated Texture Extraction from Multiple Images to Support Site Model Refinement and Visualization," *Proc. Computer Graphics and Visualization*, Plzen, Czech Republic, 1996.