Site Model Acquisition under the UMass RADIUS Project*

Robert T. Collins, Allen R. Hanson, Edward M. Riseman

Department of Computer Science
Lederle Graduate Research Center
Box 34610, University of Massachusetts
Amherst, MA. 01003-4610

Abstract

A set of image understanding (IU) modules is being developed for performing several geometric site modeling tasks, including initial model acquisition, model extension, model-to-image registration and site model refinement. This paper describes how the UMass system would acquire an initial site model. IU algorithms have been developed to hypothesize potential building roofs in an image, automatically locate supporting geometric evidence in other images, and determine the precise shape and position of the new buildings via multi-image triangulation. This process is demonstrated on a subset of images from the RADIUS Model Board 1 data set.

1 Introduction

The University of Massachusetts is developing a set of image understanding modules for automated site model acquisition, extension and refinement as part of the ARPA/ORD RADIUS project. This paper focuses on algorithms for automated building model acquisition. These algorithms are presented by way of an experimental case study using images J1–J8 of the RADIUS Model Board 1 data set. In this experiment, 25 building models were generated, covering a large portion of the model board site. The study was conducted in order to exercise and evaluate current model acquisition procedures on a realistic task.

There are many stages in the model acquisition process. This paper steps through the following subtasks:

1. line segment extraction
2. camera resection
3. building detection
4. multi-image epipolar matching
5. constrained, multi-image triangulation, and
6. projective intensity mapping.

*This work was funded by the RADIUS project under ARPA/Army TEC contract number DACA76-92-C-0041 and by ARPA/TACOM contract DAAE07-91-C-R035.

Description of each task will follow a standard pattern. First, a statement of task motivation and goals is presented. Second, a brief overview of the algorithm currently being used to perform the task is given. Detailed algorithmic descriptions are outside the scope of this paper, and will be provided elsewhere. Last, results from the Model Board 1 site modeling experiment are presented. The goal is to present a fair evaluation of current performance by showing representative successes, failures, and a quantitative analysis of results.

Buildings come in all sizes and shapes. To maintain a tractable goal for our research efforts we have chosen initially to focus on a single generic class of building models, namely flat-roofed, rectilinear structures. The simplest example of this class is a rectangular box-shape; however other examples include L-shapes, U-shapes, and indeed any arbitrary building shape such that pairs of adjacent roof edges are perpendicular and lie in a single plane. The most prevalent building types not included in this class are peaked-roof structures. Expanding current algorithms to deal with peaked roofs is a priority for the next stage of system development.

This paper ends with a sketch of how the model acquisition process described here fits within a larger site modeling framework being developed at UMass. In the near future we plan to evaluate model extension and refinement techniques using the detailed site model acquired in this experiment.

2 Radius Model Board 1

The model acquisition experiment used as a running example throughout this paper was 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. Ground truth measurements are 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

Figure 1: A sample image from Model Board 1

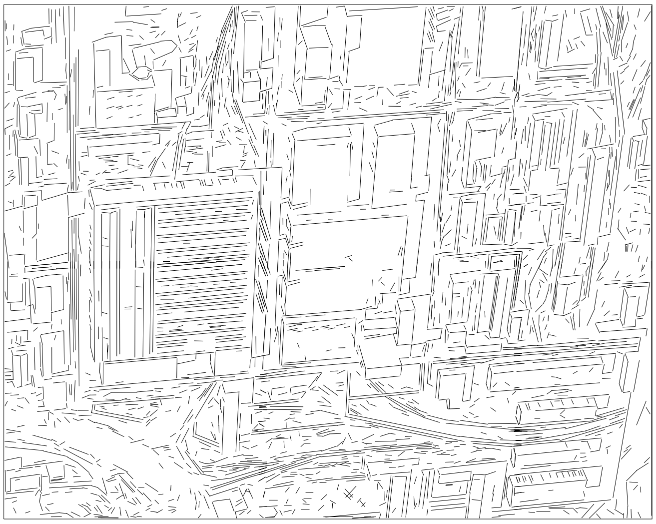

Figure 2: Line segments extracted from Figure 1

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 (i.e. $\times 500$) object coordinates). Each image contains approximately 1320×1035 pixels, with about 11 bits of grey level information per pixel. The dimensions of each image vary slightly because the images have been resampled and subjected to unmodeled geometric and photometric distortions that simulate actual operating conditions. A later set of undistorted images was provided, which we plan to use for model refinement.

3 Model Acquisition Tasks

3.1 Line Segment Extraction

Motivation. 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. Many algorithms for acquiring building models rely on extracted straight line segments.

Algorithm. We use the Boldt algorithm for extracting line segments [2]. At the heart of the Boldt algorithm is a hierarchical grouping system inspired by the Gestalt laws of perceptual organization. Zero-crossing points 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. Each iteration results in a set of increasingly longer line segments.

Our current implementation of the Boldt algorithm cannot handle full-sized 1320×1035 images. For this reason, the following procedure was performed for each image J1–J8. First, image resolution was reduced by half using Gaussian filtering and subsampling. The reduced image was then cut into overlapping subimages that were processed separately by the Boldt line extraction algorithm. All line segments found were translated and scaled back into the original image coordinate system, and filtered so that all line segments in the final set had a length of at least 10 pixels long and a contrast of at least 15 grey levels.

Results. This procedure produced roughly 2800 line segments per image. Figure 2 shows a representative set of lines, extracted from the image shown in Figure 1. Breaking each image into overlapping pieces introduced some artifacts into the line data. In particular, lines are fragmented at subimage boundaries, and lines lying totally within an overlapping area are duplicated. No attempt was made to post-process the line data to remove these artifacts, and the performance of subsequent algorithms did not appear to be degraded.

3.2 Camera Resection

Motivation. Camera resection (calibration) is a precursor for many site modeling tasks. Algorithms for camera resection traditionally use a set of 3D-to-2D feature correspondences to solve for the internal (lens) and external (pose) parameters of the camera for each image, but we use the term in an extended manner to describe any process that determines the projective relationship between image and scene, or between images. All of the algorithms discussed in this paper represent camera parameters using a 3x4 projective transformation matrix (sometimes called a Direct Linear Transform or DLT matrix). This representation makes no distinction between internal and external parameters.

Algorithm. Ideally, images to be used for site modeling purposes would be resected prior to the application of image understanding modules for automated building acquisition. Indeed, that is the goal of the upcoming ARPA/ORD Model Supported Positioning (MSP) project. The model board images were not supplied with an accurate set of camera parameters, however.

We originally formed DLT matrices for images J1–J8 using the resected camera parameters provided with version 1.0 of the RCDE (RADIUS Common Development Environment) software package [6]. The RCDE camera parameters worked fine for building detection and epipolar matching, but the building triangulation results were not very accurate when compared with corresponding 3D ground truth measurements. An investigation into the cause showed that the RCDE resections used a set of incorrectly measured ground truth points that was distributed with an early version of the Model Board 1 data set. The faulty resections will be corrected in version 2.0 of the RCDE.

To get more accurate triangulation results, we resected the images ourselves by directly estimating the 11 free parameters of the DLT matrix for each image. Matrix elements were computed by setting the lower right-hand element of the DLT matrix to 1, then estimating the remaining elements using an iterative least squares procedure to minimize the sum of squared residual errors between projected ground truth points (the correct ones) and their hand-selected image locations.

Results. Table 1 shows the average residual error for the DLT resections we performed. The residual error for each image is in the 2-3 pixel range, representing the level of unmodeled geometric distortion present in each image. Since the ground scale distance is 18 inches, this corresponds to a backprojection error of roughly 3–4.5 feet in object space. This is a significant amount of error, and presents a good test of system robustness. As mentioned earlier, model refinement procedures will later be applied using an undistorted set of images.

Table 1: RMS errors (in pixels) for J1–J8 resections.

image number	J1	J2	J3	J4
RMS error	1.95	1.93	2.72	2.38
image number	J5	J6	J7	J8
RMS error	2.25	2.87	2.38	2.04

3.3 Building Detection

Motivation. The goal of automated building detection is to roughly delineate building boundaries that will later be verified in other images by epipolar feature matching and triangulated to create 3D geometric building models.

Algorithm. The building detection algorithm is based on finding image polygons corresponding to the boundaries of flat, rectilinear rooftops in the scene. The algorithm is described in detail elsewhere in these proceedings [4]. Briefly, possible roof corners are identified by convolution with a set of oriented corner templates that respond to perspective projections of flat, orthogonal rooftop corners in the scene. Perceptually compatible corner pairs initiate a search for supporting line segment data. All corners and supporting lines are entered into a feature-relation graph and 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; virtual corner features may be hypothesized to complete a cycle, if necessary. Rooftops are finally extracted by a graph-theoretic algorithm that partitions the feature-relation graph into a set of maximally weighted, independent cycles representing closed, high-confidence building roofs.

Results. The building detector was run on image J3. This happens to be a near-nadir view, but nothing in the code precludes using one of the oblique views instead (see [4]). Roof detection is computationally expensive due to low-level feature extraction and the rapid growth of the feature-relation graph with image size. For this experiment the image was partitioned into nine separate chunks, loosely representing different "functional areas". To further speed up processing time, only templates for finding corners oriented with respect to the predominant N-S, E-W grid plan of the scene were used.

The roof detector generated 40 polygonal rooftop hypotheses. Most of the hypothesized roofs are rectangular, but six are L-shaped. Outlines of the extracted rooftops are shown in Figure 3. Alphabetic labels key into the discussion below. First, 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 3) 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 3). Note that the correct outer

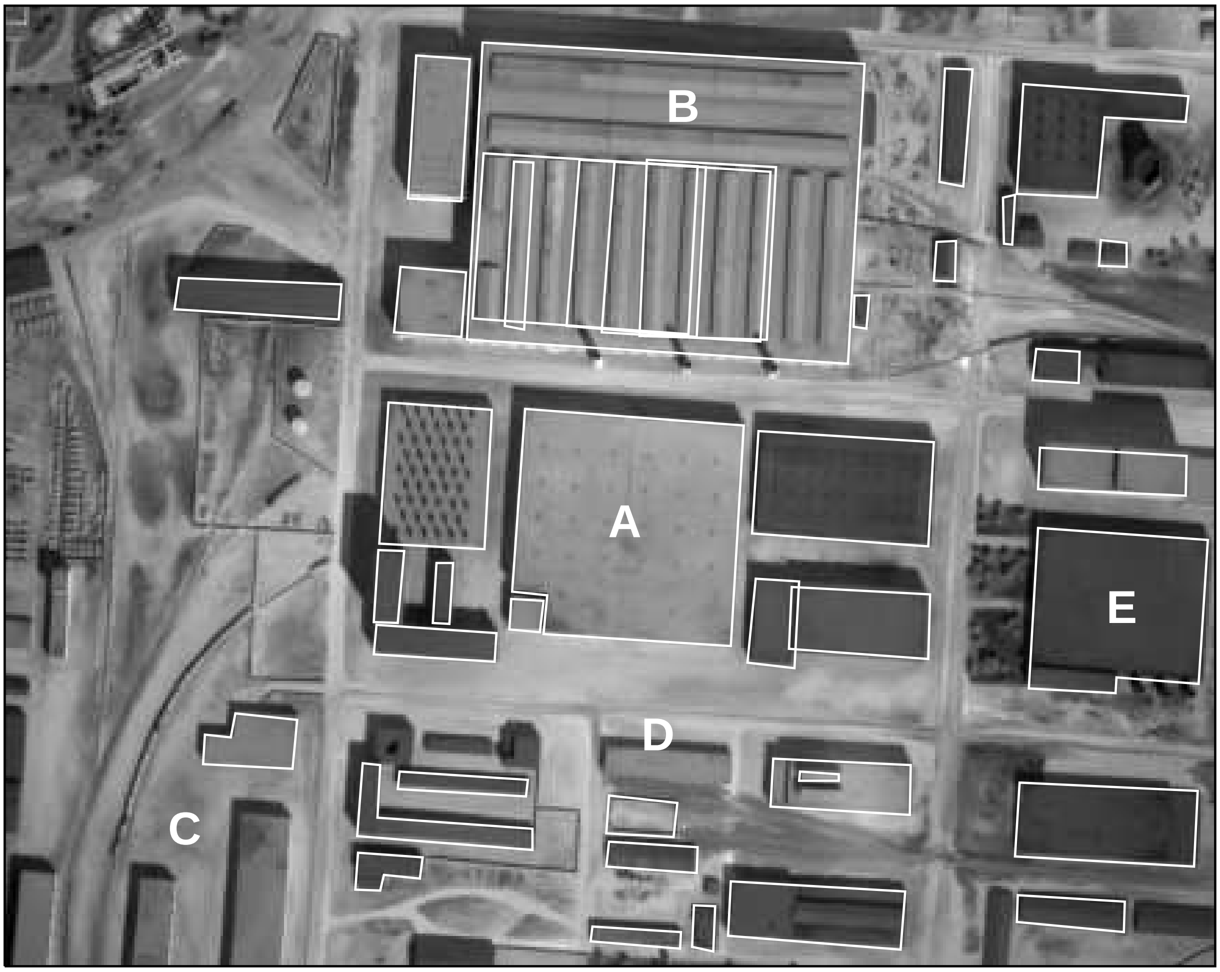

Figure 3: Roof hypotheses extracted from image J3. Alphabetic labels are referred to in the text.

outline of this building roof is detected, however. The set of parallel 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.

There are also some false negatives, which are buildings that should have been detected, but weren't. 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. The current system is built implicitly around the idea of detecting complete building models; partial building structure information that is extracted is not carried along. Although the subsequent epipolar feature matching and multi-image line triangulation routines are already able to handle such building "fragments", additional code would be necessary to merge the partial building wireframes produced from different images into a single building model.

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 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.

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 (e.g. **E**) are outlined with single polygons rather than with one polygon per roof level. Some peaked roof buildings were also outlined, even though they do not conform to the generic assumptions underlying the system.

3.4 Multi-image Epipolar Matching

Motivation. 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.

Algorithm. The key problem in epipolar matching is disambiguation of multiple potential matches. One way to avoid ambiguity is to match higher-level structures that are more distinctive. Direct implementation of this apprach is problematic, however, since failure to extract the high-level structure in another image will cause a failure to find a match, even when partial low-level evidence for the matching structure is available.

We match rooftop polygons by searching for each component line segment separately and then fusing the results. For each polygon segment from one image, an appropriate epipolar search area is formed in each of the other images, based on the known camera parameters (resected DLT matrices) and the assumption that the roof is flat. This quadrilateral search area is scanned for possible matching edges, each potential match implying a different roof height in the scene via a simple cross ratio calculation. Results from each line search are combined in a 1-dimensional histogram, each potential match voting for a particular roof height. Each vote is weighted by compatibility of the match in terms of expected line segment orientation and length. A single global histogram accumulates height votes from multiple images, and for multiple edges in a rooftop polygon. After all votes have been tallied, the histogram bucket containing the most votes yields an estimate of the roof height in the scene and a set of correspondences between rooftop edges and image line segments from multiple views.

Results. For the Model Board 1 experiment, the minimum and maximum values for the epipolar height histogram were chosen based on the range of Z-coordinates present in the set of measured ground truth points. The histogram contained 24 buckets with a height range of roughly 12 feet per bucket. After epipolar voting was completed for a rooftop polygon, correspondences were 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. This criterion was chosen because the 3D line triangulation algorithm will fail to converge in this case. Based on this criterion, epipolar matching failed on eight rooftop polygons. Six were either peaked or multi-layer roofs that did not fit the generic flat-roofed building assumption, and the other two were building fragments with some sides shorter than the minimum length threshold on the line segment data.

At this stage 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 3. The sixth was the fenced in area appearing directly below label **D** in that image. We believe that pointing to building hypotheses that are presented by the system to either accept or reject them is an acceptable level of interaction when creating a new site model. However, we are actively investigating methods for detecting and removing such mistakes automatically.

3.5 Multi-image Line Triangulation

Motivation. 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 are imposed for more reliable results.

Algorithm. We have implemented a constrained, nonlinear estimation algorithm for simultaneous multi-image, multi-line triangulation of 3D line structures with object-level constraints. 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 edge are the Plücker coordinates of the algebraic 3D line coinciding with the edge – specific points of interest, like vertices of the rooftop polygon, are computed as the intersections of these infinite algebraic lines. Plücker coordinates are a way of embedding the 4-dimensional manifold of 3D lines into R^6. Each line is represented by a pair of 3-vectors $(\boldsymbol{a}, \boldsymbol{b})$ such that $\boldsymbol{a} \cdot \boldsymbol{a} = 1$ and $\boldsymbol{a} \cdot \boldsymbol{b} = 0$. Vector $\boldsymbol{a}$ is the unit orientation vector of the line, and $\boldsymbol{b}$ is the moment vector of the line about the origin (it is normal to the plane containing both the line and the origin, with length equal to the distance of the line from the origin). Although the Plücker representation requires 6 parameters to be estimated for each line rather than 4, it simplifies the representation of geometric constraints between lines. For the generic flat-roofed rectilinear building class being considered here, we specify a set of constraints to ensure that pairs of adjacent lines in a traversal around the polygon are perpendicular, that all lines are coplanar, and that all lines are perpendicular to the Z-axis of the local site coordinate system. These conditions are linear and quadratic constraints when represented as functions of the Plücker coordinates.

An iterative, nonlinear least-squares procedure determines the Plücker coordinates for all lines simul-

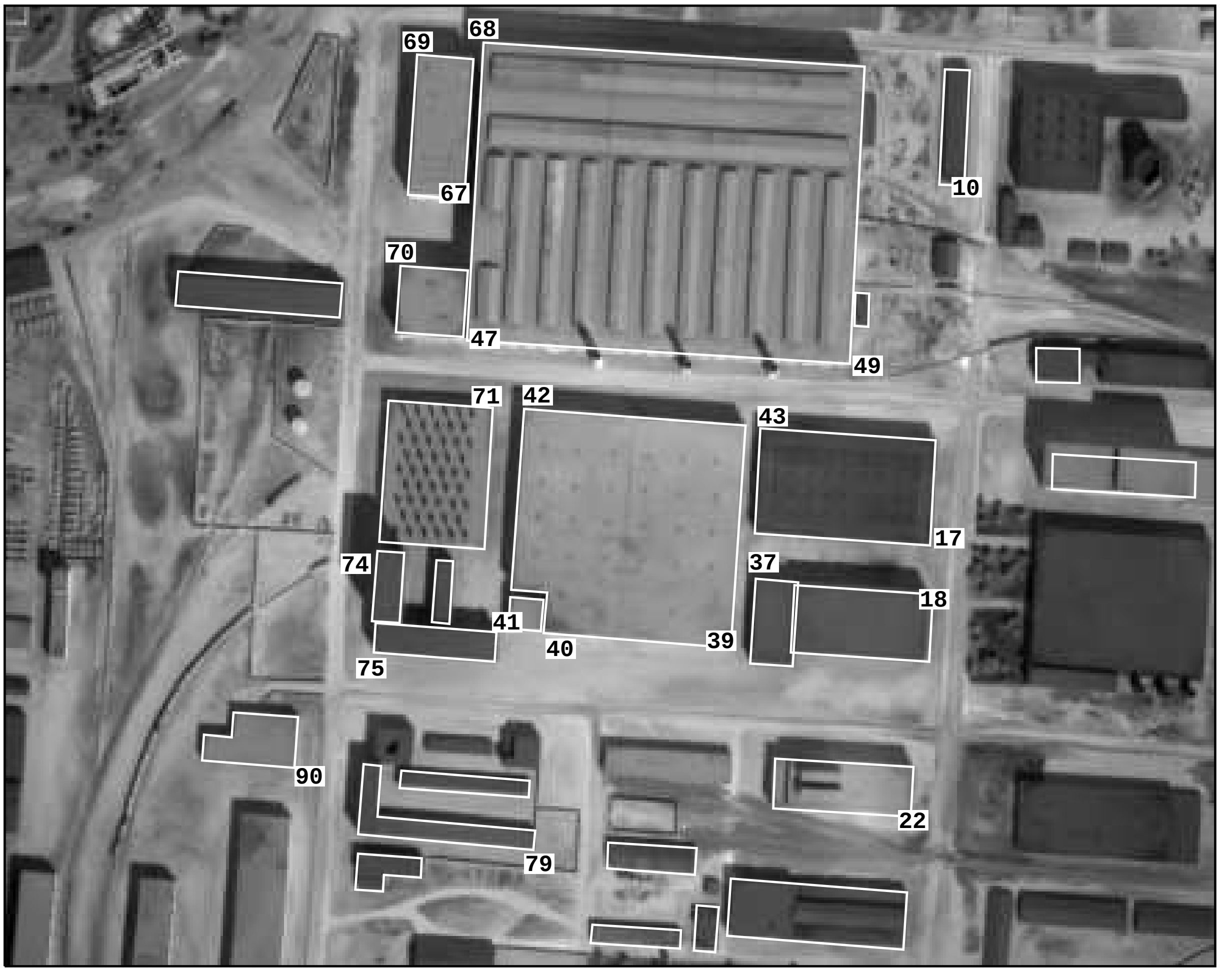

Figure 4: Reprojection of 3D triangulated rooftops back into image J3 (compare with Figure 3.) Numeric labels mark 21 roof vertices where ground truth measurements are known.

taneously such that all the object-level constraints are satisfied and an objective "fit" function is minimized that measures how well each projected algebraic line aligns with the 2D image segments that correspond to it. A number of different objective measures are being considered; the current one is a function of the sum of squared distances from each projected infinite line to the endpoints of corresponding 2D line segments in the image. Nonlinear estimation algorithms typically require an initial estimate that is then iteratively refined. We used the original rooftop polygon extracted by the building detector, and the roof height estimate computed by the epipolar matching algorithm, to generate an initial, flat, 3D roof polygon.

After triangulation, each 3D rooftop polygon is extruded down to the ground to form a volumetric model. For the Model Board 1 site we represented the ground as a horizontal plane with Z-coordinate value determined from the ground truth measurements. More generally, we will soon be combining our symbolic building extraction routines with the digital terrain maps produced by the UMass Terrain Reconstruction System [7].

Results. Outlines of the final set of triangulated rooftops are shown in Figure 4. 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 of these split-level buildings were automatically filtered out during epipolar matching, but this one managed to survive. Determining how to automatically detect and remove such errors is an ongoing research issue – there is information contained in the epipolar histograms and triangulation residuals that has yet to be taken advantage of.

To evaluate the 3D accuracy of the triangulated building polygons, 21 roof vertices were identified where ground truth measurements are known. These locations are labeled in Figure 4 with numeric indices that are keyed to the file of Model Board 1 ground truth measurements. Table 2 shows the Euclidean distances between triangulated polygon vertices and their ground truth locations The average distance is 4.31 feet, which is reasonable given the level of geometric distortion present in the images (see Section 3.2).

Table 2: Euclidean distance (in feet) between triangulated and ground truth building vertex positions. Numeric indices correspond to the labeled positions in Figure 4.

index	error	index	error	index	error
10	6.21	**41**	3.53	**69**	2.78
17	1.20	**42**	5.21	**70**	2.12
18	13.70	**43**	4.70	**71**	2.62
22	3.41	**47**	3.88	**74**	2.62
37	6.75	**49**	4.22	**75**	4.87
39	3.59	**67**	3.85	**79**	2.30
40	4.30	**68**	4.18	**90**	4.58

It is instructive to decompose the distance error into its horizontal and vertical components. 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.

Also note that the error associated with point 18 appears to be an outlier – it is twice as large as the next largest distance. The building was not triangulated well, due in part to its extremely close proximity to a neighboring building, which interferes with correct matching and triangulation. It is no coincidence that the vertex error computed for the neighboring building is the second largest error.

3.6 Projective Intensity Mapping

Motivation. Projective mapping of image intensities (rendering) onto polygonal building model faces enhances their visual realism and provides a convenient storage mechanism for later symbolic extraction of detailed surface structure.

Algorithm. Planar projective transformations provide a mathematical description of how surface structure from a planar building facet maps into an image. By inverting this transformation using known building position and camera DLT matrices, intensity information from each image can be back-projected to "paint" the walls and roof of the building model. This is performed for multiple images, leading to a library of intensity maps for all building facets, under a variety of viewing conditions.

By storing surface information with the object, intensity mapping provides a convenient storage method for later symbolic extraction of detailed surface structures like windows, doors and roof vents. Furthermore, this subsequent processing becomes greatly simplified. For example, rectangular lattices of windows or roof vents can be searched for in the unwarped intensity maps without complication from the effects of perspective distortion. Secondly, specific surface structure extraction techniques can be applied only where relevant, i.e. window and door extraction can be focused on building wall intensity maps, while roof vent computations are performed only on roofs. This is one component of an extended effort on our part towards automatic recognition of general object classes without requiring significant effort by the user, e.g. recognizing classes of doors, windows, etc., and eventually vehicles, roads, and most of the object types expected in these domains.

Results. For each of the 25 volumetric building models, 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. The best intensity map for each facet in terms of resolution and contrast was chosen and stored with the model. Figure 5 shows an example of the intensity information stored with each building model. Since multiple images are used, intensity information from all faces is available even though they are not all visible from any single view.

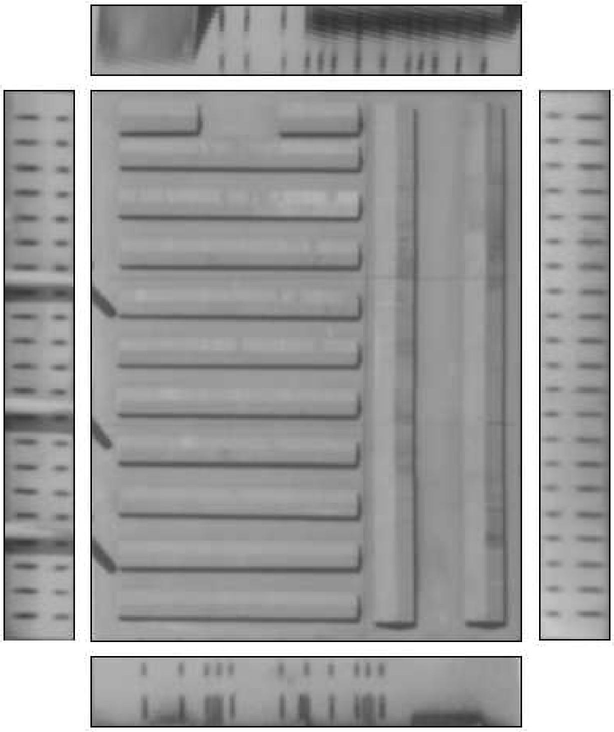

Figure 5: Intensity map information is stored with the planar facets of a building model.

The intensity-mapped building models are being used to construct a graphical site model that can be examined interactively on an SGI workstation. A simulated video "fly-through" of the site is also being produced, to demonstrate the level of realism achievable by these modeling techniques, and to investigate the use of visualization techniques for interactive evaluation of modeling results. Future work will be directed towards combining intensity information from multiple views of each polygonal building facet to remove visual artifacts caused by shadows and occlusion and to potentially increase the clarity of the surface intensity maps using super-resolution fusion techniques.

4 Conclusion

A set of IU algorithms for automated site model acquisition was presented. The algorithms currently assume a generic class of flat roofed, rectilinear buildings. When run on image J3 of the Model Board 1 imagery, an automated building detector produced 40 rooftop hypotheses. Supporting evidence was located in other images via epipolar line segment matching, and the precise 3D shape and location of each building was determined by constrained multi-image line triangulation. Through a process of filtering and attrition, we ended up with 25 building models that represent most of the central buildings in the site. Projective mapping of intensity information from the images onto these polyhedral models results in a compelling site model display that can be interactively explored on the SGI using fly-through graphics.

The algorithms described here are part of a larger system being developed at UMass for site modeling applications [3]. The UMass design philosophy emphasizes model-directed processing, rigorous 3D perspective camera equations, and fusion of information across multiple images for increased accuracy and reliability. Acquired site models will be used for automated model-to-image registration and resection of new images [1]. Proper registration between an incoming image and a stored geometric site model determines the position and appearance of model features in the image. The model can then be overlaid on the image to aid visual change detection and verification of expected scene features. Two other important site modeling tasks are *model extension* – updating the geometric site model by adding or removing new buildings based on the results of change detection – and *model refinement* – iteratively refining the shape, placement and surface structure of building models as more views become available [5]. Model extension and refinement are expected to be ongoing processes that are repeated whenever new images become available, each updated model becoming the current site model for the next iteration. Thus, over time, the site model will be steadily improved to become more complete and more accurate.

5 Acknowledgements

This paper would not be possible without the Radius team members: Yong-Qing Cheng, Chris Jaynes, Frank Stolle, and Xiaoguang "XG" Wang, and the software support and wizardry of Robert Heller and Jonathan Lim.

References

[1] J. Beveridge and E. Riseman, "Hybrid Weak-Perspective and Full-Perspective Matching," *Proceedings IEEE Computer Vision and Pattern Recognition*, Champaign, IL, 1992, pp. 432–438.

[2] M. Boldt, R. Weiss and E. Riseman, "Token-Based Extraction of Straight Lines," *IEEE Transactions on Systems, Man and Cybernetics,* Vol. 19, No. 6, 1989, pp. 1581–1594.

[3] R. Collins, A. Hanson, E. Riseman and Y. Cheng, "Model Matching and Extension for Automated 3D Site Modeling," *Proceedings Arpa Image Understanding Workshop,* Washington, DC, April 1993, pp. 197–203.

[4] C. Jaynes, F. Stolle and R. Collins, "Task Driven Perceptual Organization for Extraction of Rooftop Polygons," *Proc. Arpa Image Understanding Workshop,* 1994 (these proceedings).

[5] R. Kumar and A. Hanson, "Application of Pose Determination Techniques to Model Extension and Refinement," *Proceedings Darpa Image Understanding Workshop*, San Diego, CA, January 1992, pp. 727–744.

[6] Martin Marietta and SRI International, *RCDE User's Guide,* Martin Marietta, Management and Data Systems, Philadelphia, PA, 1993.

[7] H. Schultz, "Terrain Reconstruction from Oblique Views," *Proc. Arpa Image Understanding Workshop,* 1994 (these proceedings).