Collected University of Massachusetts RADIUS Papers

Y. Cheng, R. Collins, C. Jaynes, A. Hanson
E. Riseman, H. Schultz, F. Stolle, X. Wang

CMPSCI Technical Report 95-43

May, 1995

The following is a list of original citations for the papers that are collected in this technical report:

R. Collins, C. Jaynes, F. Stolle, X. Wang, Y. Cheng, A. Hanson and E. Riseman, "A System for Automated Site Model Acquisition," Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision II, SPIE Proceedings Vol. 7617, Orlando, FL, April 1995, (proceedings to appear).

R. Collins, A. Hanson and E. Riseman, "Site Model Acquisition under the UMass RADIUS Project," Arpa Image Understanding Workshop, Monterey, CA, November 1994, pp. 351-358.

R. Collins, Y. Cheng, C. Jaynes, F. Stolle, X. Wang, A. Hanson and E. Riseman, "Site Model Acquisition and Extension from Aerial Images," International Conference on Computer Vision, Cambridge, MA, June 1995, (to appear).

H. Schultz, "Terrain Reconstruction from Oblique Views," Arpa Image Understanding Workshop, Monterey, CA, November 1994, pp. 1001-1008.

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, December 1994, pp. 152-159.

Y. Cheng, R. Collins, A. Hanson and E. Riseman, "Triangulation without Correspondences," Arpa Image Understanding Workshop, Monterey, CA, November 1994, pp. 993-1000.

Abstract

This collection of six papers represents a snapshot of current work at the University of Massachusetts on automated acquisition, extraction, and refinement of geometric site models under the ARPA/ORD RADIUS program. The first paper "A System for Automated Site Model Acquisition" introduces the problem of geometric site modeling from aerial images, briefly discusses the general system requirements for automating the site model construction process, and outlines the key algorithms making up a system for extracting a restricted class of buildings from multiple aerial images. The second paper "Site Model Acquisition under the UMass RADIUS Project" further describes the system being developed and provides additional results on the RADIUS modelboard imagery. The third paper "Site Model Acquisition and Extension from Aerial Images" includes a description of the site model extension techniques being developed. The first three papers overlap to some degree, but vary in their emphasis on the different system components and results.

The fourth paper "Terrain Reconstruction from Oblique Views" describes a stereo algorithm for the extraction of a digital elevation map from two or more views of a site. The algorithm has been designed to work effectively even when the views are widely separated and highly oblique. Work is in progress on integrating the geometric site model with the digital elevation map. The fifth paper "Task Driven Perceptual Organization for Extraction of Rooftop Polygons" is an in-depth description of the perceptual grouping and organization algorithm developed for extracting rooftops from a monocular image. These rooftop hypotheses are then verified across multiple images and their 3D structure determined by means of a multi-image optimization procedure, as described in the first three papers. The final paper "Triangulation Without Correspondence" discusses two algorithms for reconstructing three dimensional points from two sets of noisy two-dimensional points without a-priori knowledge of the point correspondences between the sets.

A system for automated site model acquisition*

Robert T. Collins, Chris Jaynes, Frank Stolle, Xiaoguang Wang,
Yong-Qing Cheng, Allen R. Hanson, Edward M. Riseman

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

ABSTRACT

A system has been developed to acquire, extend and refine 3D geometric site models from aerial imagery. The system hypothesizes potential building roofs in an image, automatically locates supporting geometric evidence in other images, and determines the precise shape and position of the new buildings via multi-image triangulation. Projectively warped image intensity maps are associated with the faces of each recovered building, allowing realistic rendering of the scene from new viewpoints.

Keywords: aerial image understanding, building extraction, 3D site modeling

1 INTRODUCTION

Acquisition of 3D geometric site models from aerial imagery is currently the subject of an intense research effort in the U.S., sparked in part by the ARPA/ORD RADIUS project.[7,8,5,13] We have developed a set of image understanding modules to acquire, extend and refine 3D volumetric building models. 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 both man-made and natural features of interest, and to determine their 3D shape and placement in the scene. This paper focuses on algorithms for automatically extracting models of buildings. 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,[4] allowing the model to 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 features,[6] and *model refinement* – iteratively refining the shape and placement of features as more views become available. Model extension and refinement are 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 is steadily improved to become more complete and more accurate.

*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.

UMass has designed and implemented a system for automatically extracting building from multiple, overlapping images of a site. To maintain a tractable goal for our research efforts, we have chosen initially to focus on a single generic class of buildings, 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 horizontal plane. The system is designed to operate over multiple images exhibiting a wide variety of viewing angles and sun conditions. The system is designed to perform well at one end of a data-vs-control complexity spectrum, namely a large amount of data and a simple control structure, versus the alternative of using less data but more complicated processing strategies. In particular, while the system can be applied to a single stereo pair, it generally performs better (in terms of number of buildings found) when more images are used.

Section 2 begins with a specification of general input requirements of the UMass system. This is followed in Section 3 by a breakdown of the system into its key algorithmic components: 1) line segment feature extraction, 2) monocular building rooftop detection, 3) multi-image epipolar rooftop matching, 4) multi-image wireframe triangulation, and 5) projective intensity mapping. This paper concludes with a brief summary and a statement of future work.

2 General system requirements

The UMass building extraction system was developed on a Sun Sparc 10, using the Radius Common Development Environment (RCDE).[12] The RCDE is a combined Lisp/C++ system that supports the development of image understanding algorithms for constructing and using site models. In particular, the RCDE provides a convenient framework for representing and manipulating images, camera models, object models and terrain models, and for keeping track of their various coordinate systems, inter-object relationships, and transformation/projection equations. The RCDE also provides utilities for interactively developing site models, specifying tie points, and for performing photo-resection.

2.1 Images

Acquisition of a 3D site model requires a set of overlapping images of the site. The UMass system is designed to operate over multiple images, typically five or more, exhibiting a wide variety of viewing angles and sun conditions. The number five is chosen arbitrarily to allow one nadir view plus four oblique views from each of four perpendicular directions (e.g. North, South, East and West). This configuration is not a requirement, however. Indeed, some useful portions of the system require only a single image, namely line segment extraction and building rooftop detection. On the other hand, epipolar rooftop matching and wireframe triangulation require, by definition, at least two images, with robustness and accuracy increasing when more views are available. Once again, the number five has been chosen arbitrarily, and perhaps only three well-chosen images would suffice, but verification of this is a matter for further experimentation.

Although best results require the use of many images with overlapping coverage, the system allows considerable freedom in the choice of images to use. Unlike most other building extraction systems, this system does not currently use shadow information, and works best if used on images with different sun angles, or with no strong shadows at all. Also, the term "epipolar" as used here does not imply that images need to be in scan-line epipolar alignment, as required by many traditional stereo techniques. The term is used instead in its general sense as a set of geometric constraints imposed on potentially corresponding image features by the relative orientation of their respective cameras. The relative orientation of any pair of images is computed from the absolute orientation of each individual image (see Section 2.3).

2.2 Site coordinate system

Reconstructed building models are represented in a local site coordinate system that must be defined prior to the reconstruction process. The system assumes this is a "local-vertical" Euclidean Coordinate System, that is, a Cartesian X-Y-Z coordinate system with its origin located within or close-to the site, and the positive Z-axis facing upwards (parallel to gravity). The system can be either right-handed or left-handed. Under a local-vertical coordinate system, the Z values of reconstructed points represent their vertical position or "height" in the scene, and X-Y coordinates represent their horizontal location in the site.

2.3 Camera models

For each image given to the system, the absolute orientation of the camera with respect to the local site coordinate system must be known. This includes both the internal orientation (lens/digitizer parameters) and the external orientation (pose parameters) of the camera. Given the absolute orientation for each image, the system computes all the necessary relative orientation information needed for determining the epipolar geometry between images. Camera models can be specified in two ways. For the **perspective frame camera model**, absolute orientation for each camera is supplied as a 3×4 projective transformation matrix describing (in homogeneous coordinates) how points in the site coordinate system project into points in the image coordinate system. This simple representation makes no distinction between internal and external camera parameters. Translation between "standard" photogrammetric parameterizations (e.g. focal length, principle point coordinates, camera location vector and rotation Euler angles) and the 3×4 matrix representation is provided by the RCDE.

Many aerial photographs, particularly satellite images, are generated by nontraditional imaging systems for which the standard perspective frame camera model is not an adequate description. The **fast block interpolation projection** (FBIP) camera model has been proposed as an alternative description of the imaging process in these situations. The general idea is to break space into "blocks" and then generate local frame camera approximations within each block in such a way that adjacent frame approximations agree at the block boundary, in a manner somewhat analogous to approximating a nonlinear function by a piecewise linear one. This representation easily handles 2D image nonlinearities such as camera lens distortion, as well as 3D space nonlinearities caused by the refraction of light through layers of the atmosphere.

Integrating the FBIP camera model into image understanding algorithms is potentially tricky, since it violates the fundamental assumption underlying most work with traditional, perspective camera models, namely the assumption that straight lines in the world will appear straight in the image. The FBIP camera model not only raises representational concerns such as whether the edge of a building in the image can be adequately characterized by a single straight line segment, but also strikes at a deeper level, invalidating such fundamental geometric notions as vanishing points and epipolar geometry. Our interpretation of FBIP camera model is that it is possible to derive a local 3×4 projective transformation matrix that provides an accurate approximation to the imaging process within a given 3D region of interest spanning the spatial extents of a single building.

2.4 Digital terrain map

Currently, the UMass system explicitly reconstructs only the rooftops of building structures, and relies on vertical extrusion to form a volumetric 3D wireframe model of the whole building. In other words, perpendiculars are dropped from each corner of the reconstructed building rooftop down to the ground, and connected by a building base formed as a vertical translation of a copy of the roof polygon. The extrusion process relies on knowing the local terrain, namely the ground height (Z value) at each location in the scene. We assume this information is represented as an array of elevations, or in the special case of flat ground planes as a horizontal plane equation $Z = z_0$. Representation of digital terrain maps in either format, along with their use in providing a basic ground level for vertical extrusion, is supported by the RCDE. Future versions of the system will use digital terrain maps automatically extracted from stereo image pairs (nadir or oblique) by a correlation-based

terrain reconstruction system developed recently at UMass. The technical details of that system also appear in these processings.[14]

2.5 Other required parameters

In addition to the general information described above, a few miscellaneous parameters and thresholds are required to be supplied by the user before the system can be run. The most important of these are:

- **max-building-height** – the maximum possible height of any building that will be included in the site model. This threshold is used to limit the extent of epipolar search regions. The lower this threshold can be, the smaller the search area for rooftop feature matches will be, leading to faster searches with higher likelihood of finding the correct matches.
- **min-building-width** – the minimum horizontal extent (width or length) of any building that will be included in the site model. This is, loosely speaking, a way of specifying the desired "resolution" of the resulting site model, since any buildings having horizontal edges shorter than this threshold will probably not be found. Setting this value to a relatively long length essentially ensures that only large buildings in the site will be modelled.

3 Algorithmic building blocks

The UMass building extraction system currently follows a simple processing strategy. To acquire a new site model, an automated building detector is run on one image to hypothesize potential building rooftops. Supporting evidence is the located in other images via epipolar line segment matching, and the precise 3D shape and location of each building is determined by multi-image triangulation and extrusion. Image intensity information is backprojected onto each face of these polyhedral building models, to facilitate realistic rendering from new views.

This section outlines the key algorithms that together comprise the UMass building extraction system. These algorithms are: line segment extraction, building rooftop detection, epipolar rooftop matching, multi-image wire-frame triangulation, and projective intensity mapping. The description of these algorithms is illustrated with sample results from two sites, the Schenectady County Air National Guard base (Figure 1), and Radius Model Board 1 (Figure 2).

3.1 Line segment extraction

To help bridge the huge representational gap between pixels and site models, a straight line feature extraction routine is applied to produce a set of symbolic line segments, representing geometric image features of potential interest such as building roof edges. We use the Boldt algorithm for extracting line segments.[3] 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 (difference in average intensity level across the line) values are similar. Each iteration results in a set of increasingly longer line segments. The final set of line segment features (Figures 3 and 4) can be filtered according to length and contrast values supplied by the user.

Although the Boldt algorithm does not rely on any particular camera model, the utility of extracting straight lines as a relevant representation of image/scene structure is based on the assumption that straight lines in the world (such as building edges) will appear reasonably straight in the image. To the extent that this assumption remains true at the scale of the objects being considered, such as over a region of the image containing a single

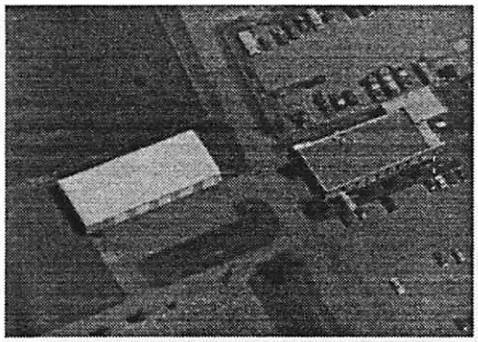

Figure 1: Sample subimage from Schenectady dataset.

Figure 2: Sample subimage Radius Model Board 1.

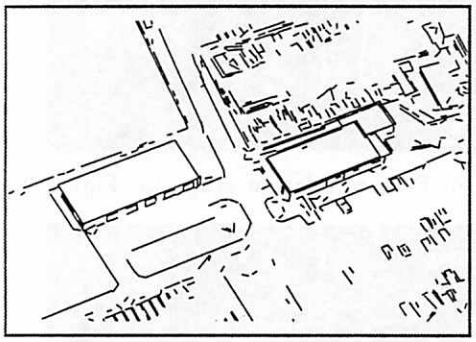

Figure 3: Boldt lines for Figure 1.

Figure 4: Boldt lines for Figure 2.

building, then straight line extraction remains a viable feature detection method. However, very long lines spanning a significant extent of the image, such as the edges of airport runways, may become fragmented depending on the amount of curvature introduced into the image by nonlinearities in the imaging process.

3.2 Building rooftop detection

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. The UMass building detection algorithm[9] is based on perceptual grouping of line segments into image polygons corresponding to the boundaries of flat, rectilinear rooftops in the scene. Perceptual organization is a powerful method for locating and extracting scene structure. The rooftop extraction algorithm proceeds in three steps; low level feature extraction, collated feature detection, and hypothesis arbitration. Each module generates features that are used at during the next phase and interacts with lower level modules through top-down feature extraction.

Low level features in this system are straight line segments and corners. The domain assumption of flat-roofed rectilinear structures implies that rooftop polygons will be produced by flat horizontal surfaces with orthogonal corners. Orthogonal corners in the world are not necessarily orthogonal in the image, however. To determine a set of relevant corner hypotheses, pairs of line segments with spatially proximate endpoints are grouped together into candidate image corner features. Each potential image corner is then backprojected into a nominal Z-plane in the scene, and that hypothetical *scene corner* is tested for orthogonality.

Mid-level collated features are sequences of perceptually grouped corners and lines that form a chain (Figures 5 and 6). A valid chain group must contain an alternation of corners and lines, and can be of any length. Chains are a generalization of the collated features in earlier work[8] and allow final polygons of arbitrary rectilinear shape to be constructed from low level features. Collated feature chains are represented by paths in a feature

relation graph. Low level features (corners and line segments) are nodes in the graph, and perceptual grouping relations between these features are represented by edges in the graph. Nodes have a certainty measure that represents the confidence of the low level feature extraction routines; edges are weighted with the certainty of the grouping that the edge represents. A chain of collated features inherits an accumulated certainty measure from all the nodes and edges along its path.

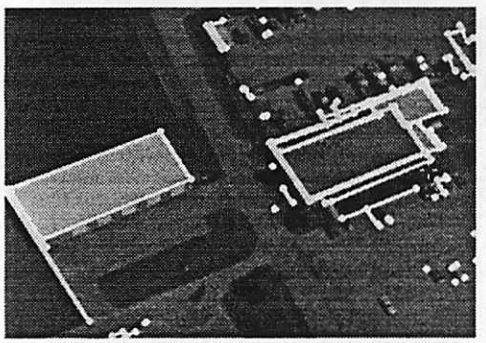

Figure 5: Feature relation graph for Figure 1.

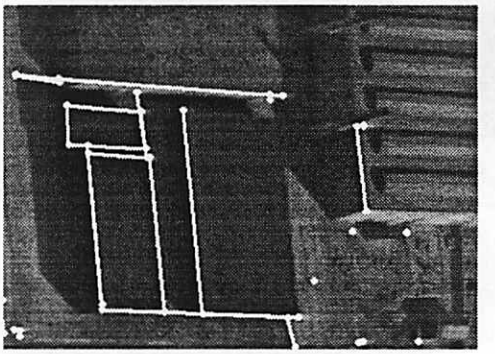

Figure 6: Feature relation graph for Figure 2.

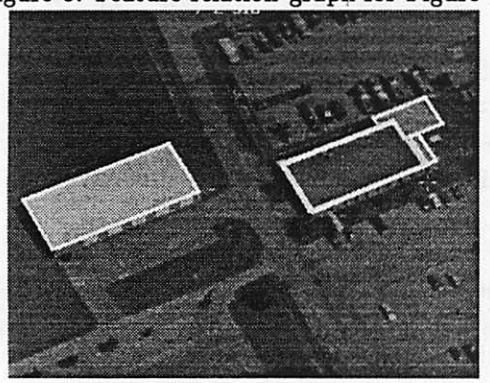

Figure 7: Final rooftop hypotheses for Figure 1.

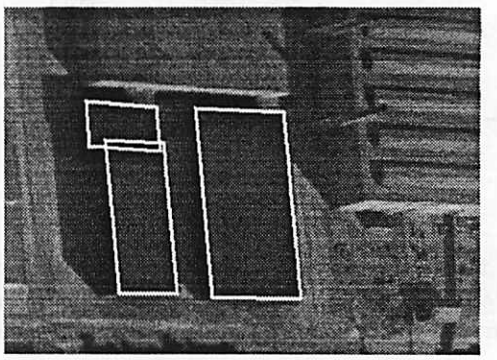

Figure 8: Final rooftop hypotheses for Figure 2.

High level polygon hypothesis extraction proceeds in two steps. First, all possible polygons are computed from the collated features. Then, polygon hypotheses are arbitrated in order to arrive at a final set of non-conflicting, high confidence rooftop polygons (Figures 7 and 8). Polygon hypotheses are simply closed chains, which can be found as cycles in the feature relation graph. All of the cycles in the feature relation graph are searched for in a depth first manner, and stored in a dependency graph where nodes represent complete cycles (rooftop hypotheses). Nodes in the dependency graph contain the certainty of the cycle that the node represents. An edge between two nodes in the dependency graph is created when cycles have low level features in common. The final set of non-overlapping rooftop polygons is the set of nodes in the dependency graph that are both independent (have no edges in common) and are of maximum certainty. Standard graph-theoretic techniques are employed to discover the maximally-weighted set of independent cycles is, which is output by the algorithm as a set of independent high confidence rooftop polygons.

While searching for closed cycles, the collated feature detector may be invoked in order to attempt closure of chains that are missing a particular feature (an example occurs in Figure 6). The system then searches for evidence in the image that such a virtual feature can be hypothesized. In this way, the rooftop detection process does not have to rely on the original set of features that were extracted from the image. Rather, as evidence for a polygon accumulates, tailor-made searches for lower level features can be performed. This type of top-down inquiry increases system robustness.

3.3 Epipolar line segment 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. The primary difficulty to be overcome during epipolar matching is the resolution of ambiguous potential matches, and this ambiguity is highest when only a single pair of images are used. For example, the epipolar search region for a roof edge match will often contain multiple potentally matching line segments of the appropriate length and orientation, one of which comes from the corresponding roof edge, but the others coming from the base of the building, the shadow edge of the building on the ground, or from roof/base/shadow edges of adjacent buildings (see Figure 9). This situation is exacerbated when the roof edge being searched for happens to be nearly aligned with an epipolar line in the second image. The resolution of this potential ambiguity is the reason that simultaneous processing of multiple images with a variety of viewpoints and sun angles is preferred in the UMass system.

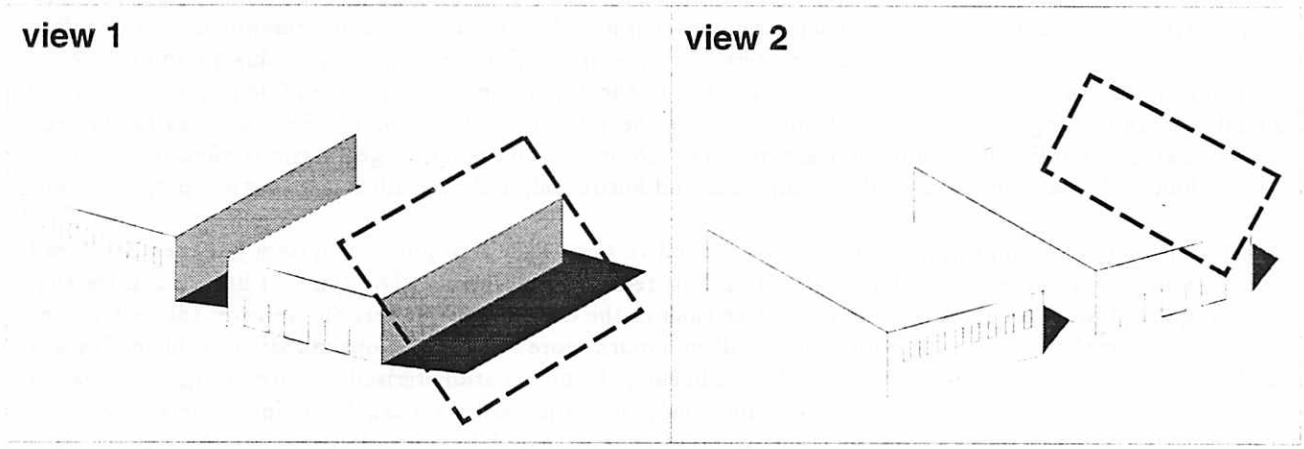

Figure 9: Multiple ambiguous matches can often be resolved by consulting a new view.

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 geometry and the assumption that the roof is flat. This quadrilateral search area is scanned for possible matching edges, the disparity of each potential match implying a different roof height in the scene. 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. This allows for correct handling of fragmented line data, for example, since the combined votes of all subpieces of a fragmented line count the same as the vote of a full-sized, unfragmented line. 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.

3.4 Wireframe triangulation/extrusion

After finding a set of rooftop edge correspondences via epipolar matching, multi-image triangulation is performed to determine the precise size, shape, and position of the roof polygon in the local 3D site coordinate system. A nonlinear estimation algorithm has been developed for simultaneous multi-image, multi-line triangulation of 3D line structures.

Two versions of the triangulation subsystem have been developed. In the first, 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. 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, a set of constraints is specified 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. An iterative, nonlinear least-squares procedure determines the Plücker coordinates for all lines simultaneously 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.

Although triangulation of line structures via Plücker coordinates is general, in the sense that any set of 3D lines can be represented, we have found this approach to be computationally burdensome and numerically unstable. The reason for this is mainly due to the number of parameters in the representation and the number of constraints that must be imposed to achieve a unique, geometrically accurate solution. In particular, triangulation of a rooftop polygon containing n lines requires $6 \times n$ parameters to represent the Plücker coordinates, plus an addition $2 \times n$ Lagrange multipliers to ensure a unique solution (recall that the dimension of the line manifold is 4, thus $6-4=2$ additional constraints are required for each line to make the solution vector unique). Further constraints (and thus more Lagrange multiplier parameters) are necessary to impose the required geometric configuration on the lines in the final polygon, namely that all are coplanar and horizontal, and that adjacent pairs are perpendicular.

In response to these computational difficulties, a second version of the triangulation system has been developed using a specialized parameterization for representing flat, rectilinear polygons. The types of line structures that can be triangulated are considerably more restrictive than in the earlier, general version, however the restrictions mesh well with current system assumptions and result in a much more streamlined optimization problem. Instead of each line being represented separately, a whole rectilinear polygon is parameterized at once, using the variables shown in Figure 10. 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 parameterθ. Finally, each separate line within the polygon is represented by a single value r_i representing the signed perpendicular distance of that line from some nominal point in the plane, usually chosen to be near the center of mass of the polygon being estimated. The representation is simple and compact, and the method of Lagrange multipliers is no longer necessary since the coplanarity and rectilinearity constraints on the polygon's shape are already built in to the representation.

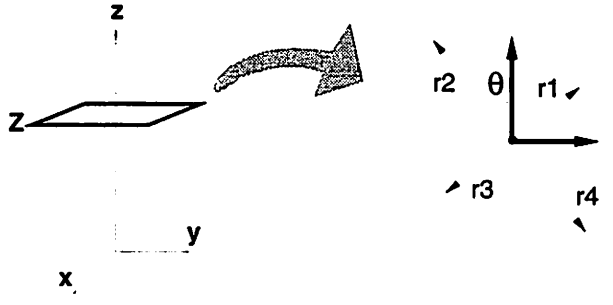

Figure 10: Parameterization of a flat, rectilinear polygon for multi-image triangulation.

Regardless of which parameterization is chosen, nonlinear estimation algorithms typically require an initial estimate that is then iteratively refined. In this system, the original rooftop polygon extracted by the building detector, and the roof height estimate computed by the epipolar matching algorithm, are used to generate an initial, flat, roof polygon. After triangulation, each 3D rooftop polygon is extruded down to the ground, as determined by the digital terrain map for the site (see Section 2.4), to form a volumetric wireframe model.

3.5 Projective intensity mapping

To provide added realism for visual displays, and as a convenient means of storage for later detailed processing of building surface information, mechanisms have been developed for projectively warping image intensities onto polygonal building facets. 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 geometry, intensity information from each image can be backprojected to "paint" the walls and roof of the building model. Since multiple images are used, intensity information from all faces of the building polygon can be recovered, even though they are not all seen in any single image (see Figure 11). The full intensity-mapped site model can then be rendered to predict how the scene will appear from a new view (Figure 12), and on high-end workstations realistic real-time "fly-throughs" can be generated. For more details on the construction of the site model used to generate Figure 12, see (Collins, 1994).

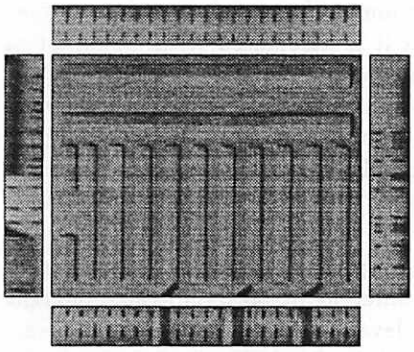

Figure 11: Intensity maps are stored with the planar facets of a building model.

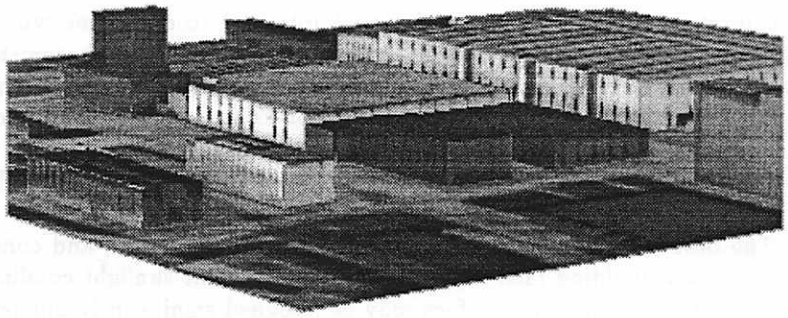

Figure 12: Intensity-mapped site model rendered from a new view.

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.

When processing multiple overlapping images, each building facet will often be seen in more than one image, under a variety of viewing angles and illumination conditions. This has led to the development of a systematic mechanism for managing intensity map data, called the Orthographic Facet Library. The orthographic facet library is an indexed data set storing all of the intensity-mapped images of all the polygonal building facets that have been recovered from the site. Usually, a horizontal roof facet appears in all the aerial site images and thus has a complete set of intensity-map versions in the library. Vertical wall facets usually show up only in a subset of the site images, however, so fewer intensity-map versions are available to choose from. Each intensity-map version is tagged with a variety of spatial and photometric indices (e.g. viewing angle, resolution, sun angle) in order to facilitate retrieval and analysis by image understanding algorithms. As intensity-mapped building facets accumulate in the facet image library, knowledge about the site improves; albeit in an implicit, image-based form.

When using the facet library to render a new view of the site, it is necessary to distill the information contained in multiple intensity-mapped versions of each building facet into a single "best" image representation for that facet. Two alternative solutions have been tried so far. The first approach is to use the pixels in the **best representative version** of each facet to paint the given surface. The "goodness" of an image with respect to a

particular building facet is based on a heuristic measure that takes into account the camera viewing angle, the sun angle, and the placement and geometry of other buildings in the site, all of which allow the system to compute the size, relative orientation, and photometric contrast of the facet in the image, as well as predict the percentage of the facet covered by shadows or occlusion in that view. The advantage of best version representation is its simplicity, in that only a heuristic function is calculated for each view and no further image processing is needed. The drawback of this method is that sometimes occlusions or shadows appear in ever image of a building facet, thus the representative will have to include those artifacts no matter which image is chosen. The best version representation was used to render the building in Figure 11.

In contrast to the best version approach, the **best representative piece** method takes occlusions and shadows into account. As intensity-map versions are placed in the library, pixels in the facet are partitioned into "pieces" according to whether they are sunlit or in shadow. Pixels that are labeled as occluded areas are discarded and are not considered to be a part of any piece. The idea of the best piece representation is to assign a heuristic value to each piece of an intensity-map version, rather than to the entire version. When rendering a new view, each pixel on a building's surface is backprojected to determine which pieces it is associated with. This set of pieces is ordered according o their heuristic values, and the photometric value for the pixel is selected from the highest-rated piece. Hence, all the pixels in the rendered image are the best ones available. Note, however, that some pixels in the rendered image might not exist in any of the pieces in the library, when they correspond to portions of building that have never been seen in any of the images. These pixels are painted black by default. The best version representation was used to render the site model in Figure 12.

The best piece representation is a method of data fusion, and compatability problems arise in that different pieces of each building face can appear under different sunlight conditions in different images, and thus different portions of the same building face may be assigned significantly different grey-levels, leading to a patchy appearance. One reasonable way to solve this problem is to make all the versions of the facet "similar" in intensity. Currently, a simple histogram adjustment technique is used to make the intensity distributions of all the pieces associated with a single building face uniform with respect to each other. The biggest sunlit piece of the facet is chosen as the model piece against which all other pieces are transformed.

4 SUMMARY AND FUTURE WORK

UMass has developed an image understanding system for automated site model acquisition. The algorithms currently assume a generic class of flat roofed, rectilinear buildings. To acquire a new site model, an automated building detector is run on one image to hypothesize potential building rooftops. Supporting evidence is located in other images via epipolar line segment matching, and the precise 3D shape and location of each building is determine by multi-image triangulation. Projective mapping of image intensity information onto these polyhedral building models results in a realistic site model that can be rendered using virtual "fly-through" graphics. In an operational scenario, this process would be repeated as new images become available, gradually accumulating evidence over time to make the site model database more complete and more accurate.

Several avenues for system improvement are open. One high priority is to add capabilities for detecting and triangulating peaked roof buildings. Another significant improvement would be extending the epipolar matching and triangulation portions of the system to analyze why a particular building roof hypothesis failed to be verified. There are many cases where the rooftop detector has outlined split-level buildings with a single roof polygon; automatic detection of these situations, followed by splitting of the rooftop hypothesis into two separate hypotheses, would result in an improvement in system performance.

These symbolic building extraction procedures will soon be combined with a correlation-based terrain extraction system.[14] The two techniques clearly complement each other: the terrain extraction system can determine a digital elevation map upon which the volumetric building models rest, and the symbolic building extraction procedures can identify building occlusion boundaries where correlation-based terrain recovery is expected to behave poorly. A tighter coupling of the two systems, where an initial digital elevation map is used to focus

attention on distinctive humps that may be buildings, or where correlation-based reconstruction techniques are applied to building rooftop regions to identify fine surface structure like roof vents and air conditioner units, may also be investigated.

5 ACKNOWLEDGEMENTS

We would like to acknowledge the software and technical support of Robert Heller and Jonathan Lim, the video wizardry of Fred Weiss, and the administrative support of Janet Turnbull and Laurie Waskiewicz.

6 REFERENCES

[1] American Society of Photogrammetry, *Manual of Photogrammetry,* Fourth Edition, American Society of Photogrammetry, Falls Church, VA, 1980.

[2] J.R. Beveridge and E. Riseman, "Hybrid Weak-Perspective and Full-Perspective Matching," *Proc. Computer Vision and Pattern Recognition,* Champaign, IL, 1992, pp. 432-438.

[3] 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.

[4] 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.

[5] R. Collins, A. Hanson and E. Riseman, "Site Model Acquisition under the UMass RADIUS Project," *Proceedings Arpa Image Understanding Workshop,* Monterey, CA, November 1994, pp. 351–358.

[6] R. Collins, Y. Cheng, C. Jaynes, F. Stolle, X. Wang, A. Hanson and E. Riseman, "Site Model Acquisition and Extension from Aerial Images," *Proceedings IEEE International Conference on Computer Vision,* Cambridge, MA, June 1995, **to appear.**

[7] D. Gerson, 'RADIUS : The Government Viewpoint," *Proceedings of the Darpa Image Understanding Workshop,* San Diego, CA, January 1992, pp. 173–175.

[8] A. Huertas, C. Lin and R. Nevatia, "Detection of Buildings from Monocular Views of Aerial Scenes using Perceptual Grouping and Shadows," *Proc. Arpa Image Understanding Workshop,* Washington, DC, April 1993, pp. 253–260.

[9] C. Jaynes, F. Stolle and R. Collins, "Task Driven Perceptual Organization for Extraction of Rooftop Polygons," *Proceedings Arpa Image Understanding Workshop,* Monterey, CA, November 1994, pp. 359–365.

[10] R. Kumar and A. Hanson, "Robust Methods for Estimating Pose and Sensitivity Analysis," *CVGIP: Image Understanding,* Vol. 60, No. 3, November 1994, pp. 313-342.

[11] D. McKeown, "Toward Automatic Cartographic Feature Extraction," in *Mapping and Spatial Modelling for Navigation,* Nato ASI Series, Vol. F65, pp. 149–180, 1990.

[12] J. Mundy, R. Welty, L. Quam, T. Strat, W. Bremner, M. Horwedel, D. Hackett and A. Hoogs, "The RADIUS Common Development Environment," *Proceedings of the Darpa Image Understanding Workshop,* San Diego, CA, January 1992, pp. 215–226.

[13] M. Roux and D. McKeown, "Feature Matching for Building Extraction from Multiple Views," *Proceedings Arpa Image Understanding Workshop,* Monterey, CA, November 1994, pp. 331–349.

[14] H. Schultz, "Terrain Reconstruction from Widely Separated Images," *Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision II,* Spie Proceedings Vol. 7617, Orlando, FL, April 1995, **this proceedings.**

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

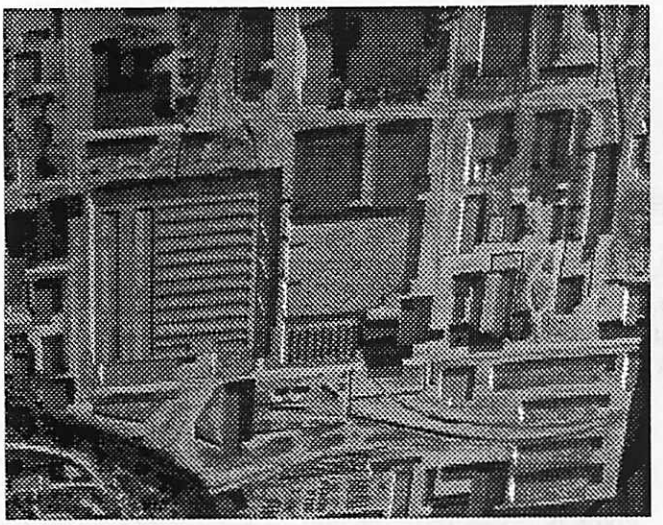

Figure 1: A sample image from Model Board 1

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 $\times$ 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 $\times$ 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

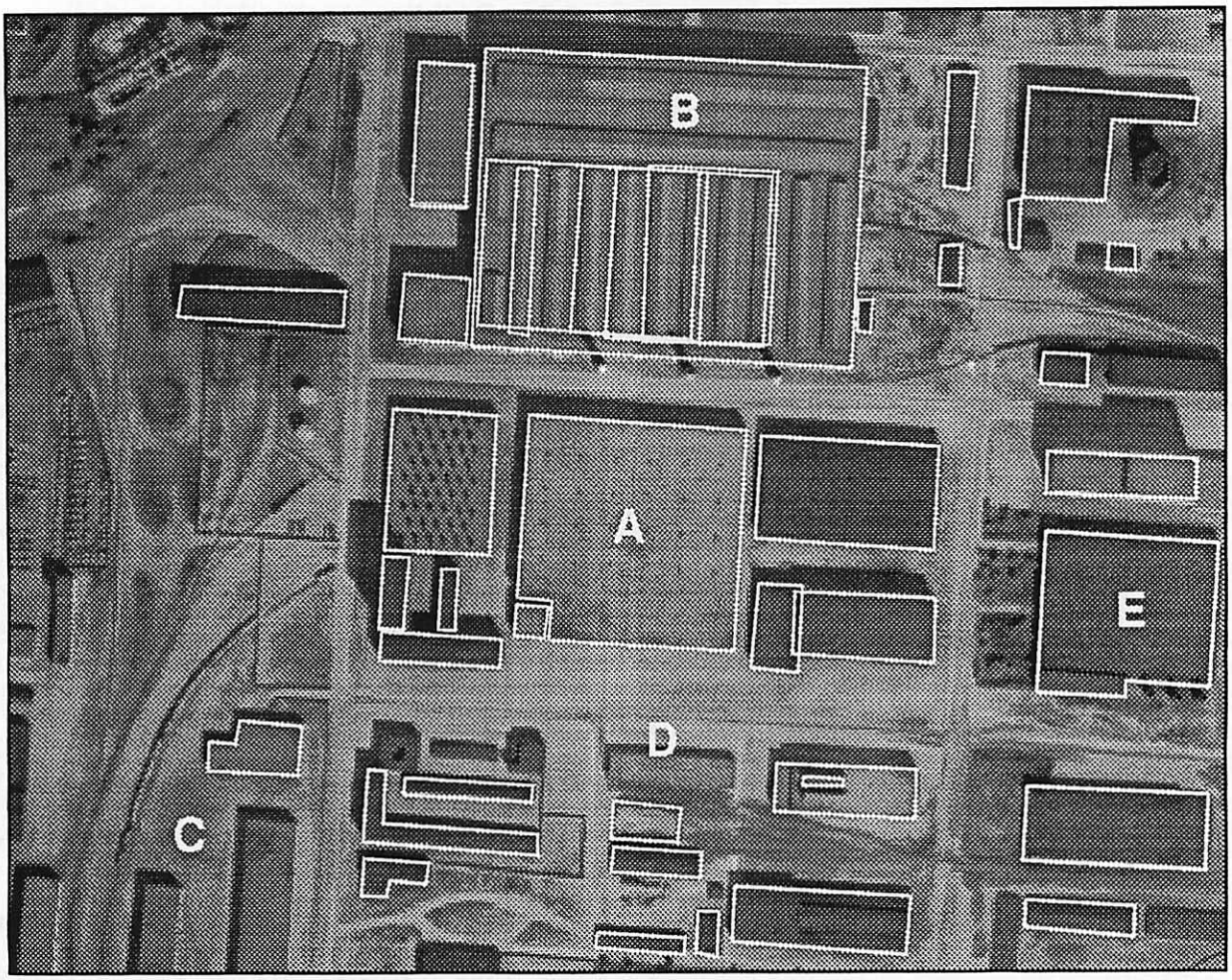

Figure 3: Roof hypotheses extracted from image J3. Alphabetic labels are referred to in the text.

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-

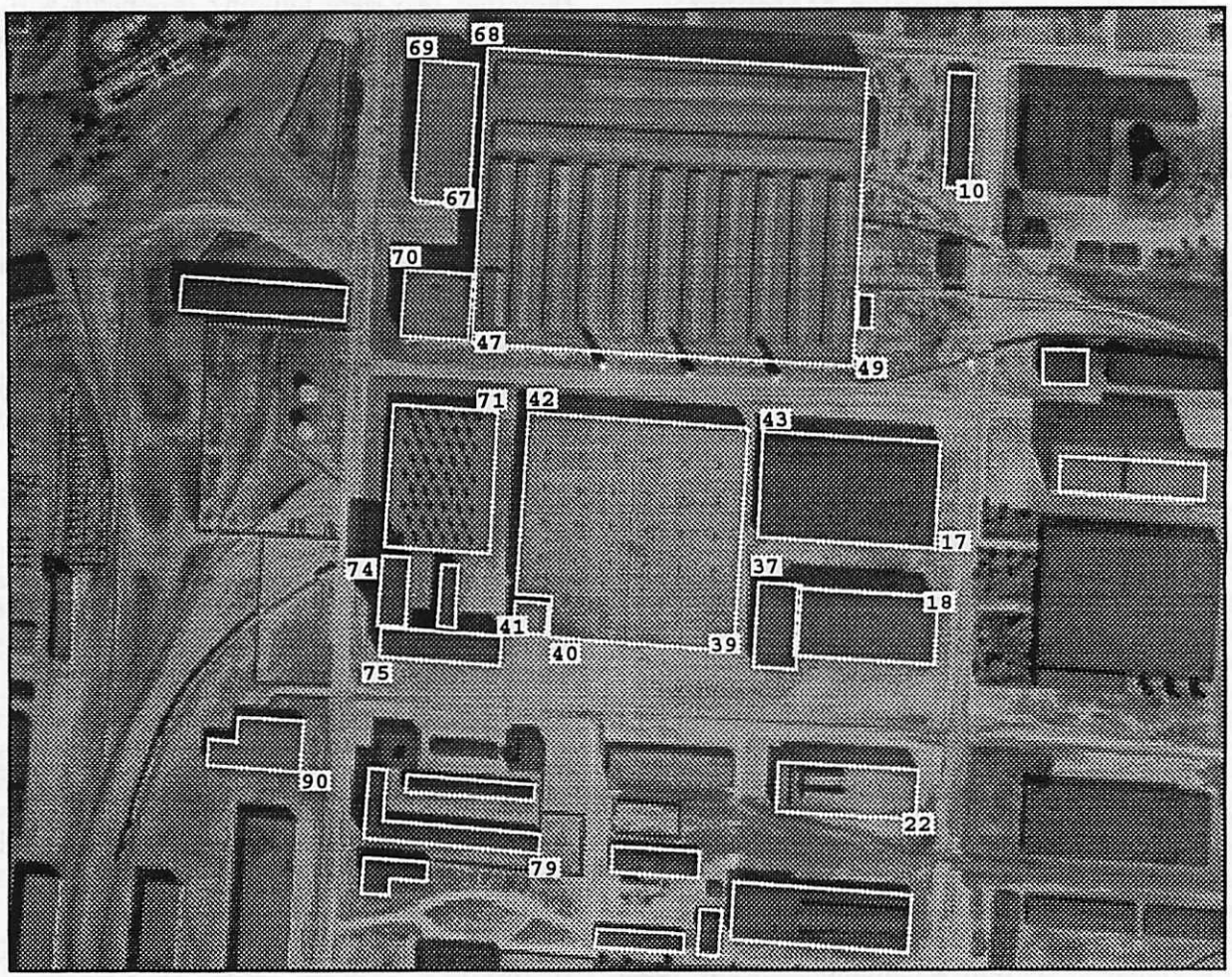

Figure 4: Reprojection of 3D triangulated rooftops back into image J3 (compare with Figure 3.) Numeric labels mark 21 roof vertices where ground truth measurements are known.

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 ver-

tices 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.

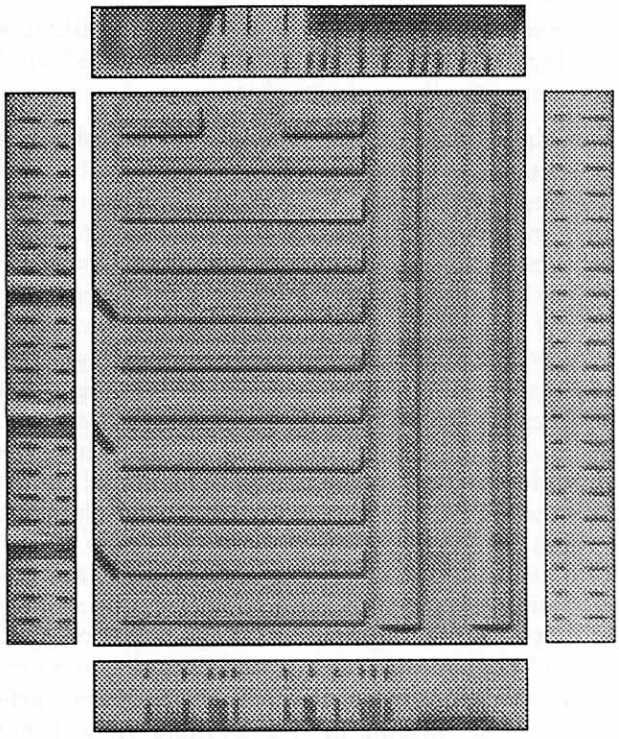

Figure 5: Intensity map information is stored with the planar facets of a building model.

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).

Site Model Acquisition and Extension from Aerial Images *

Robert T. Collins, Yong-Qing Cheng, Chris Jaynes, Frank Stolle,
Xiaoguang Wang, Allen R. Hanson, and Edward M. Riseman

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

Abstract

A system has been developed to acquire, extend and refine 3D geometric site models from aerial imagery. This system hypothesize potential building roofs in an image, automatically locates supporting geometric evidence in other images, and determines the precise shape and position of the new buildings via multi-image triangulation. Model-to-image registration techniques are applied to align new, incoming images against the site model. Model extension and refinement procedures are then performed to add previously unseen buildings and to improve the geometric accuracy of the existing 3D building models.

1 Introduction

Acquisition of 3D geometric site models from aerial imagery is currently the subject of an intense research effort, sparked in part by the ARPA/ORD RADIUS project [3, 4, 5, 8]. We have developed a set of image understanding modules to acquire, extend and refine 3D volumetric building models, and to provide a digital elevation map of the surrounding terrain. System features include 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 buildings 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 model-to-image registration of incoming images, thus allowing the model to be automatically overlaid on each image as an aid to 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 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. Model extension and refinement are 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 is steadily improved to become more complete and more accurate.

This paper focuses on algorithms for automated building model acquisition and extension. 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 horizontal plane. Acquisition of an initial site model is treated in Section 2, followed by model extension in Section 3. This paper concludes with a brief summary and a statement of future work.

2 Site Model Acquisition

The building model acquisition process involves several subtasks: 1) line segment extraction, 2) building detection, 3) multi-image epipolar matching, 4) constrained, multi-image triangulation, and 5) projective intensity mapping. These algorithms will be presented by way of an experimental case study using images J1–J8 of the RADIUS model board 1 data set. Figure 1 shows a sample image from the data set. Each image contains approximately 1320×1035 pixels, with about 11 bits of gray level information per pixel. Unmodeled geometric and photometric distortions have been added to each image to simulate actual operating conditions. The scene is a 1:500 inch scale model of an industrial site. Ground truth measurements are available for roughly 110 points scattered throughout the model, which were used to determine the exterior orientation for each image. The residual resection error for each image is in the 2–3 pixel range, representing the level of unmodeled geometric distortion present in each image. This corresponds to a backprojection error of roughly 3–4.5 feet in (simulated) object space. This is a significant amount of error that presents a good test of system robustness.

*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.

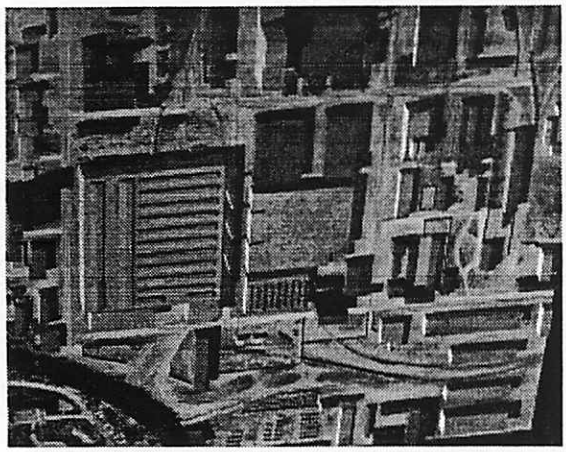

Figure 1: A sample image from Model Board 1

Figure 2: Line segments extracted from Figure 1

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 algorithms for acquiring building models rely on extracted straight line segments [2]. 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. Filtering to keep line segments with a length of at least 10 pixels and a contrast of at least 15 gray levels produced roughly 2800 line segments per image. Figure 2 shows a representative set of lines extracted from the image shown in Figure 1.

2.2 Building Detection

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. The building detection algorithm is based on finding image polygons corresponding to the boundaries of flat, rectilinear rooftops in the scene [6]. Briefly, possible roof corners are identified by line intersections. Perceptually compatible corner pairs are linked with surrounding line data, 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 partitioning the feature-relation graph into a set of maximally weighted, independent cycles representing closed, high-confidence building roofs.

Figure 3 shows the results of building detection on image J3 of the model board 1 data set. The roof detector generated 40 polygonal rooftop hypotheses. 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 Fig. 3) the segmentation is nearly perfect.

There were some false positives, i.e. 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 (area **B**) Note that the correct outer outline of this building roof is detected, however. 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 (area **C**) that are only partially in view at the edge of the image. 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. 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 (area **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.

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 line segment separately and then fusing the results. For each polygon segment from one image, an epipolar search

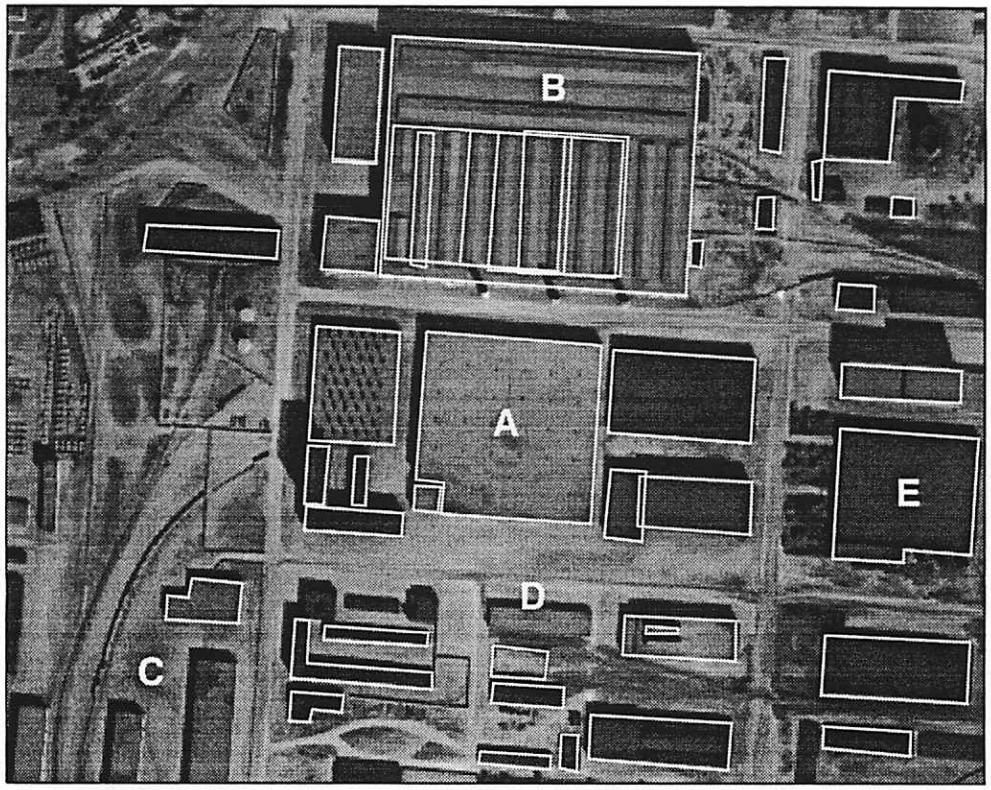

Figure 3: Roof hypotheses extracted from image J3. Alphabetic labels are referred to in the text.

area is formed in each of the other images, based on the known camera transformations 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 roof height in the scene. Results from each line search are combined in a 1-dimensional histogram, each match voting for a particular roof height, 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.

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. 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, six incorrect building hypotheses were removed by hand; detecting and removing such mistakes automatically is being actively investigated.

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. A nonlinear estimation algorithm has been developed for simultaneous multi-image, multi-line triangulation of 3D line structures. Object-space constraints are imposed for more reliable results. This algorithm is used for triangulating 3D rooftop polygons from the line segment correspondences determined by epipolar feature matching. Outlines of the final set of triangulated rooftops are shown in Figure 4.

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. 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. An iterative, nonlinear least-squares procedure determines the Plücker

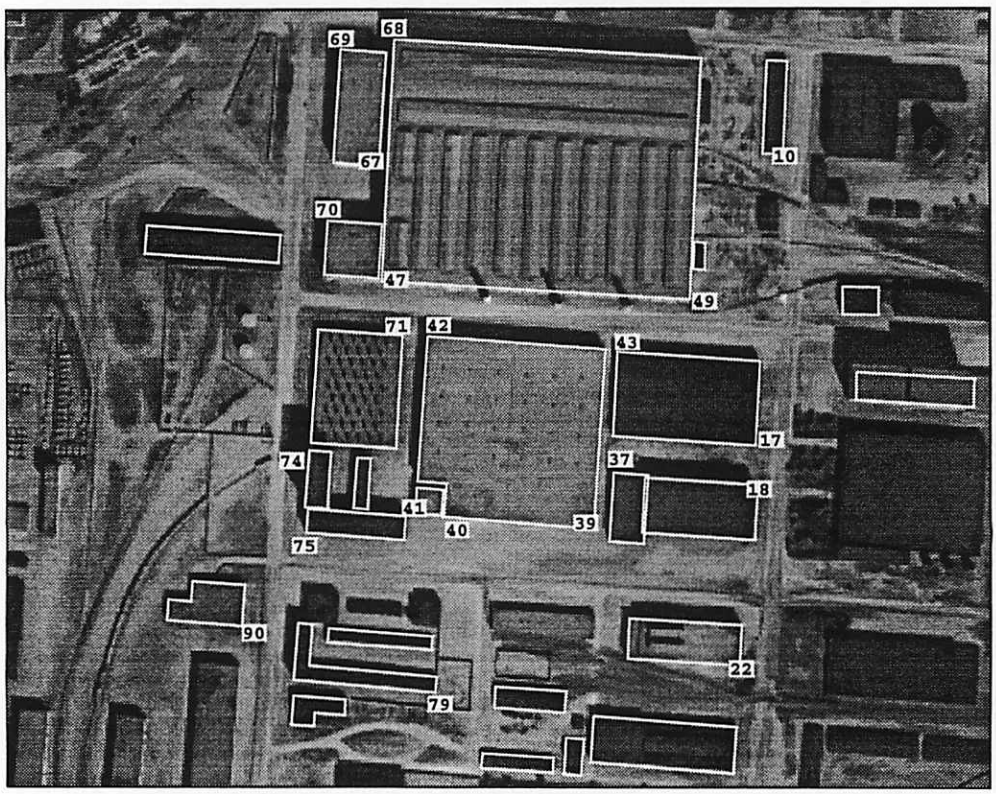

Figure 4: Reprojection of 3D triangulated rooftops back into image J3 (compare with Figure 3).

coordinates for all lines simultaneously 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.

After triangulation, each 3D rooftop polygon is 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 will soon be using digital terrain maps produced by the UMass Terrain Reconstruction System[9].

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 4). The average Euclidean distance between triangulated polygon vertices and their ground truth locations is 4.31 feet, which is reasonable given the level of 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.

2.5 Projective Intensity Mapping

Backprojection of image intensities onto polygonal building model faces enhances their visual realism and provides a convenient storage mechanism for later symbolic extraction of detailed surface structure. Planar projective transformations provide a locally valid 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 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 all faces is available, even though they are not all visible from any single view (see Figure 5). 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 "flythroughs" are achievable.

3 Site Model Extension

The goal of site model extension is to find unmodeled buildings in new images and add them into the site model database. The main difference between model extension and model acquisition is that now the 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*.

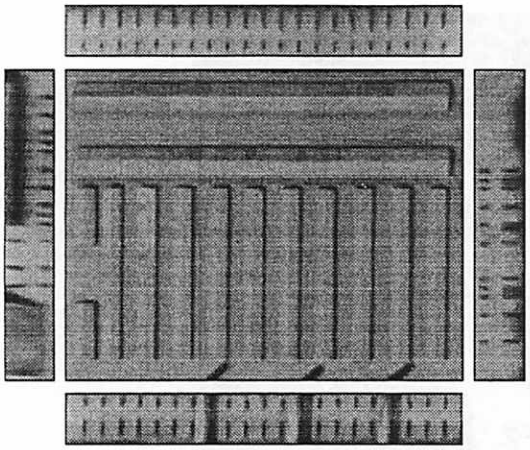

Figure 5: Intensity map information is stored with the planar facets of a building model.

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 [1] is being used. Based on a *local search* approach to combinatorial optimization, this algorithm searches the discrete space of correspondence mappings between model and image 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 needs to take into account the possibility of mistakes or *outliers* in the set of correspondences found. The robust pose estimation procedure described in [7] is being 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 projected model features fall 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.

3.1 Model Extension Example

The model extension process involves registering a current geometric site model with a new image, and then focusing on unmodeled areas to recover previously unmodeled buildings. 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 6, which shows projected building rooftops from the site model (thin) overlaid on the image. Image areas containing buildings already in the site model were masked off, and the building rooftop detector was run on the unmodeled areas. The multi-image epipolar matching and constrained multi-image triangulation procedures from Section 2 were then applied to verify the hypotheses and construct 3D volumetric building models. These were added to the site model database, to produce the extended model shown in Figure 6 (thick lines). The main reason for failure among building hypotheses that were not verified was that they represented buildings located at the periphery of the site, in an area which is not visible in very many of the eight views. If more images were used with greater site coverage, more of these buildings would have been included in the site model.

4 Summary and Future Work

A set of IU algorithms for automated site model acquisition and extension have been presented. The algorithms currently assume a generic class of flat roofed, rectilinear buildings. To acquire a new site model, an automated building detector is run on one image to hypothesize potential building rooftops. Supporting evidence is located in other images via epipolar line segment matching, and the precise 3D shape and location of each building is determine by multi-image triangulation. Projective mapping of image intensity information onto these polyhedral building models results in a realistic site model that can be rendered using virtual "fly-through" graphics. To perform model extension, the acquired site model is registered to a new image, and model acquisition procedures are focused on previously unmodeled areas. In an operational scenario, this process would be repeated as new images become available, gradually accumulating evidence over time to make the site model database more complete and more accurate.

Several avenues for system improvement are open. One high priority is to add capabilities for detecting and triangulating peaked roof buildings. Another significant improvement would be extending the epipolar matching and triangulation portions of the system to analyze why a particular building roof hypothesis failed to be verified. There are many cases where the rooftop detector has outlined split-level buildings with a single roof polygon; automatic detection of these situations, followed by splitting of the rooftop hypothesis into two separate hypotheses, would result in an improvement in system performance.

These symbolic building extraction procedures will soon be combined with a correlation-based terrain extraction system [9]. The two techniques clearly com-

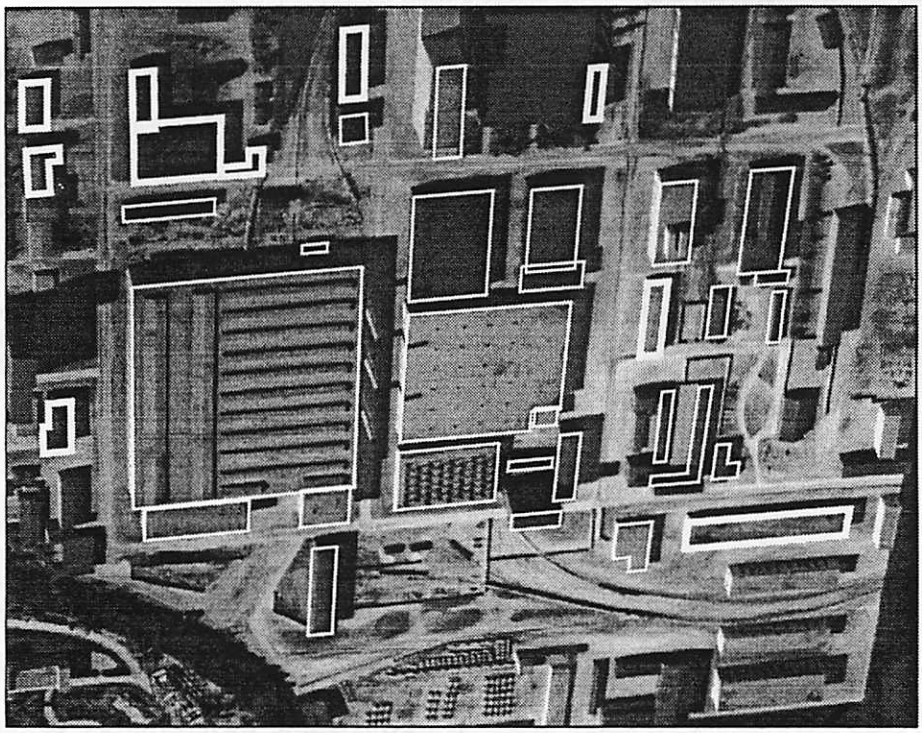

Figure 6: Updated site model projected onto image J8.

plement each other: the terrain extraction system can determine a digital elevation map upon which the volumetric building models rest, and the symbolic building extraction procedures can identify building occlusion boundaries where correlation-based terrain recovery is expected to behave poorly. A tighter coupling of the two systems, where an initial digital elevation map is used to focus attention on distinctive humps that may be buildings, or where correlation-based reconstruction techniques are applied to building rooftop regions to identify fine surface structure like roof vents and air conditioner units, may also be investigated.

Acknowledgements

We would like to acknowledge the technical and administrative support of Robert Heller, Jonathan Lim, Janet Turnbull, Laurie Waskiewicz and Fred Weiss.

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," *Trans. Systems, Man and Cybernetics*, Vol. 19(6), 1989, pp. 1581–1594.

[3] R. Collins, A. Hanson and E. Riseman, "Site Model Acquisition under the UMass RADIUS Project," *Proc. ARPA Image Understanding Workshop*, Monterey, CA, November 1994, pp. 351–358.

[4] D. Gerson, "RADIUS : The Government Viewpoint," *Proceedings DARPA Image Understanding Workshop*, San Diego, CA, January 1992, pp. 173–175.

[5] A. Huertas, C. Lin and R. Nevatia, "Detection of Buildings from Monocular Views of Aerial Scenes using Perceptual Grouping and Shadows," *Proceedings ARPA Image Understanding Workshop*, Washington, DC, April 1993, pp. 253–260.

[6] C. Jaynes, F. Stolle and R. Collins, "Task Driven Perceptual Organization for Extraction of Rooftop Polygons," *Proceedings ARPA Image Understanding Workshop*, Monterey, CA, Nov. 1994, pp. 359–365.

[7] R. Kumar and A. Hanson, Robust Methods for Estimating Pose and Sensitivity Analysis," *CVGIP: Image Understanding*, Vol. 60, No. 3, November 1994, pp. 313–342.

[8] M. Roux and D. McKeown, "Feature Matching for Building Extraction from Multiple Views," *Proceedings ARPA Image Understanding Workshop*, Monterey, CA, November 1994, pp. 331–349.

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

Terrain reconstruction from oblique views*

Howard Schultz
Department of Computer Science
University of Massachusetts, Amherst

Abstract

When a disparity map is computed from widely separated images the perspective distortion may result in a large number of false matches and poor reconstruction accuracy. This paper describes three image matching algorithms designed specifically to process images taken from widely varying viewpoints. They include a new match score, and modifications to standard subpixel and hierarchical matching techniques. The algorithms are incorporated into a stereo analysis package and the system is tested by processing a sequence of simulated images with base-to-height ratios that varied between 0.25 and 2.25, and a single pair of high altitude images with a base-to-height ratio of 0.63. Analysis of the simulated data showed that when these algorithms are implemented the reconstruction accuracy remains independent of the base-to-height ratio.

1 Introduction

For applications such as unmanned ground vehicles, stealth navigation, RADIUS, and sensor fusion, terrain maps must be reconstructed from images gathered from distant reconnaissance sources. These images present unique problems for terrain reconstruction systems because of their oblique viewing geometry and the associated large base-to-height ratios. In this paper, algorithms designed specifically to produce accurate elevation maps from image pairs with a large base-to-height ratio are discussed. It is assumed that the camera parameters and poses are known, and the discussion is focused on developing methods for computing a disparity map. The discussion is further limited to terrain in which the characteristic disparity of an object is roughly proportional to its horizontal dimensions (i.e., objects that cover a large area have large disparities and objects that cover a small area have small disparities), the surfaces are highly textured, and most surface features are visible in both images. In addition, attention is given mainly to retrieval accuracy; at this time computation speed is not considered.

When image matching is used to compute a disparity map, a dilemma occurs when the size of the correlation mask is selected. To increase robustness to random noise, the mask should be as large as possible, and to minimize the effects of projective distortion, the mask should be as small as possible [Mostafavi, 1978]. To help develop algorithms that balance these competing factors we take advantage of the fact that pixels near the mask center are less affected by projective distortion. The weighted cross-correlation match score (described in Section 2) and the subpixel image matching technique (described in Section 3) are designed specifically to place more emphasis on the pixels near the center of the correlation mask. In addition, a hierarchical matching scheme is discussed Section 4 that iteratively corrects for perspective distortion.

2 Weighted correlation mask

Starting with two views of an object (labeled R and L), the goal of image matching is to find pixel pairs (one in the R image and one in the L image) that view the same spot on the object. In the matching process a series of match scores $\rho(i, j, \delta i, \delta j), \delta i = \delta i_{min} \cdots \delta i_{max}$ and $\delta j = \delta j_{min} \cdots \delta j_{max}$ is computed between a window of pixels (correlation mask) centered at pixel $(i, j)_R$ in the R image and a similar mask centered at $(i + \delta i, j + \delta j)_L$ in the L image. By convention, disparities $(\delta i, \delta j)$ are defined relative to a fixed position in the R image, and a variable position in the L image. Next, the optimal disparity

*Sponsored by a grant from the Office of Naval Research (N00014-89-J-3229).

$(\delta i^*, \delta j^*)$ corresponding to the best match is selected. If $(\delta i^*, \delta j^*)$ satisfies the condition that $(i,j)_R$ and $(i+\delta i^*, j+\delta j^*)_L$ view the same spot on the object, the best match is said to be correct; otherwise, a false match has occurred. False matches occur when noise or distortion lowers the match score for the correct disparity and/or raises the match score for an incorrect disparity in such a way as to cause the wrong disparity to be selected.

One of the most robust and commonly used match scores is the cross-correlation coefficient $\rho(i,j,\delta i,\delta j)$ between a rectangular mask centered at $(i,j)_R$ and a similar mask centered at $(i+\delta i, j+\delta j)_L$. By definition $\rho(i,j,\delta i,\delta j)$ is given by

$$\rho(i,j,\delta i,\delta j) = \frac{Cov\left[I_R(i,j), I_L(i+\delta i, j+\delta j)\right]}{\sqrt{Var\left[I_R(i,j)\right] Var\left[I_L(i+\delta i, j+\delta j)\right]}} \quad (1)$$

where $I_R(i,j)$ and $I_L(i,j)$ are pixel intensities, $Cov\left[I_R(i,j), I_L(i+\delta i, j+\delta j)\right]$ is the covariance between masks, and $Var\left[I_R(i,j)\right]$ and $Var\left[I_L(i+\delta i, j+\delta j)\right]$ are the variances within each mask [Cochran and Medioni, 1992]. In this formula, $\rho(i,j,\delta i,\delta j)$ does not depend on the positions of the pixels within the mask, and all pixels contribute equally to the match score. Pixels near the center of the mask, however, are less affected by perspective distortions, and more emphasis should be given to these pixels. This can be done by assigning a weight to each pixel that depends on its position within the mask. Thus, the weighted average $E[I_R(i,j);A]$ of the pixel values within an arbitrarily shaped mask Q centered at pixel $(i,j)_R$ is

$$E\left[I_R(i,j);A\right] = \frac{1}{N}\sum_{i'}\sum_{j'} I_R(i',j')A(i-i',j-j')$$

where $(i',j') \in Q$, $A(i-i',j-j')$ are the mask weights that depend on the distance from summation index (i',j') to the mask center (i,j), and N is the total number of pixels within the mask. Furthermore, to ensure that the mask does not attenuate or amplify the image, the weights are normalized so that the average weight is unity.

$$1 = \frac{1}{N}\sum_{i'}\sum_{j'} A(i',j')$$

Likewise, the weighted variation at pixel $(i',j')_R$ is $I_R(i',j')A(i-i',j-j') - E\left[I_R(i,j);A\right]$, and the weighted variance $Var\left[I_R(i,j);A\right]$ and covariance $Cov\left[I_R(i,j), I_L(i+\delta i, j+\delta j)\right]$ are given by

$$Var\left[I_R(i,j);A\right] = \frac{1}{N-1}\times \sum_{i'}\sum_{j'}\left[I_R(i',j')A(i-i',j-j') - E\left[I_R(i,j);A\right]\right]^2$$

and

$$Cov\left[I_R(i,j), I_L(i+\delta i, j+\delta j);A\right] = \frac{1}{N-1}\sum_{i'}\sum_{j'} \left[I_R(i',j')A(i-i',j-j') - E\left[I_R(i,j);A\right]\right] \times \left[I_L(i'+\delta i, j'+\delta j)A(i-i',j-j') - E\left[I_L(i+\delta i, j+\delta j);A\right]\right]$$

and the weighted cross-correlation match score is given by

$$\rho(i,j,\delta i,\delta j;A) = \frac{Cov\left[I_R(i,j), I_L(i+\delta i, j+\delta j);A\right]}{\sqrt{Var\left[I_R(i,j);A\right] Var\left[I_L(i+\delta i, j+\delta j);A\right]}} \quad (2)$$

For the analyses presented in this paper two weighting functions are use, Gaussian weights given by

$$A(i-i',j-j') = \frac{2n+1}{2^{2n}}\cdot\frac{2m+1}{2^{2m}} \times \frac{2n!}{(n-i+i')!\,(n+i-i')!} \times \frac{2m!}{(m-j+j')!\,(m+j-j')!}$$

and uniform weights given by $A(i-i',j-j') = 1$, for $-n \le (i-i') \le n$ and $-m \le (j-j') \le m$. Note that when the uniform weights are used, the weighted cross-correlation match score (Equation 2) reduces to the conventional cross-correlation match score (Equation 1).

By assigning a weight of zero to all pixels that lie outside Q, the array of weighted averages $\mathbf{E}(I;A)$ for all pixels in an image may be computed by convolving I with the kernel A and dividing by N.

$$\mathbf{E}(I;A) = \frac{1}{N} I * A$$

Similarly, the computation formulas for the variance array $\mathbf{Var}(I;A)$ and covariance array $\mathbf{Cov}(I_R, S(I_L,\delta i,\delta j;A)$ are

$$\mathbf{Var}(I;A) = \frac{1}{N-1}\left[I^2 * A^2 - \frac{1}{N}(I*A)^2\right] \quad (3)$$

and

$$\mathbf{Cov}(I_R, S(I_L, \delta i, \delta j); A) = \frac{1}{N-1}\left[(I_R \cdot S(I_L, \delta i, \delta j)) * A^2 - \frac{1}{N}(I_R * A) \cdot (S(I_L, \delta i, \delta j) * A)\right]$$

where $S(I_L, \delta i, \delta j)$ is an operator that shifts an image by δi pixels in the i-dimension and δj pixels in the j-dimension. Implementation of the shift operator is important to the performance of the subpixel image matching algorithm (see Section 3 for details).

If the surface is stationary or the images are taken simultaneously, the computations may be simplified by applying epipolar constraints [Slama, 1980]. By resampling the R and L images so that each image line corresponds to an epipolar line, the vertical component of the disparity becomes identically zero, i.e., $\delta j = 0$, for all (i, j). In the following sections it is assumed that the epipolar constraints apply, thus δj is dropped from all equations and δi is replaced with δ.

3 Subpixel image matching

Reconstruction accuracy depends directly on the disparity map accuracy; therefore, significant improvements can be achieved by computing disparities to subpixel accuracy. Subpixel registration schemes rely on the assumption that near the true disparity $\tilde{\delta}$, the computed match scores are estimates of a smooth function $\tilde{\rho}(\delta)$ [Faugeras, 1993]. Thus, an estimate of the optimal disparity δ^* is found by approximating $\tilde{\rho}(\delta)$ with a model $f(\delta; c_0, c_1, \cdots)$, solving for the coefficients $c_0, c_1, \cdots$, and setting δ^* to the value of δ that optimized the model, $f'(\delta^*; c_0, c_1, \cdots) = 0$. For the subpixel matching algorithm described below, a parabolic model $f = c_0 \cdot \delta^2 + c_1 \cdot \delta + c_2$ is used; and δ^* is found by solving a least squares problem for the coefficients (c_0, c_1, c_2) and then setting $\delta^* = -2c_1/c_0$ [Tian and Huhns, 1986]. The error sources associated with this scheme are modeling error (i.e., the difference between $\tilde{\rho}(\delta)$ and $f(\delta)$) and contamination of pixel values by random noise. A detailed analysis of the effect of these error sources is beyond the scope of this paper. However, it is important to note that the effects of modeling errors and random noise become more pronounced as δ moves away from $\tilde{\delta}$ (the true disparity).

Typically, match scores are evaluated at a series of integer disparities about the previous best guess of the true disparity δ_0^*. If the interval is too narrow, an insufficient number of samples are used to estimate the location of the peak; and if the interval is too wide, the match score estimates at the ends of the interval may not be statistically significant. In either case, the location of the peak is poorly defined. For example, if $\delta = \delta_0^* + [-2, -1, 0, 1, 2]$ then only 5 observations are used to compute 3 parameters. If the range is extended to $\delta = \delta_0^* + [-4, -3, -2, -1, 0, 1, 2, 3, 4]$ the number of samples is increased to 9, but large modeling and random noise errors at the ends of the interval may contaminate the match scores used to estimate the location of the peak.

One method for solving this problem is to use a smaller disparity search range and evaluate the match scores at subpixel intervals. If the desired width of the search range is approximately ± 1.5 about the previous best guess δ_0^* and the interval between pixels is split p times, where p is an odd integer, the search range is

$$\delta_0^* - \left(\frac{3p+1}{2p}\right) + \frac{n}{p}, \quad n = 0, \cdots, 3p+1$$

For example, if $p = 5$ the disparity values are $\delta i = \delta_0^* + [-\frac{8}{5}, -\frac{7}{5}, \cdots, -\frac{1}{5}, 0, \frac{1}{5}, \cdots, \frac{7}{5}, \frac{8}{5}]$ and 17 match scores in an interval 3.2 pixels wide are used to estimate δi^*.

In the computational formulas (Equation 3) the disparities are not specified directly. Instead a subpixel shift operator $S(I_L, \delta, 0)$ is used to shift the entire image by δ pixels in the i-dimension before the convolutions with A and A^2 are computed. The shift operation is implemented by convolving I_L with an asymmetric kernel B_δ, i.e., $S(I_L, \delta, 0) = I_L * B_\delta$. For example, if $\delta = 1.2$, then $B_\delta = (0.2, 0.8, 0, 0, 0)$; and if $\delta = -0.9$, then $B_\delta = (0, 0, 0.1, 0.9, 0)$.

4 Pyramid processing

When imaging terrain it is generally true that large objects have large disparities and small objects have small disparities. When the resolution of the R and L images are reduced, smaller features disappear. Thus, only small scale disparities are lost when the low resolution images are correlated. Once the large scale disparities are recovered, the small scale disparities are recovered by processing the high resolution images and restricting the disparity

search to perturbations about the previously recovered disparities. This refinement process results in a significant reduction in the amount of computation, which in addition to saving time also reduces the chance of encountering false matches. The sequential processing from low to high resolution image pairs is referred to as hierarchical, or pyramid processing [Anandan, 1989]. Note that pyramid schemes will fail when small features have large disparities (e.g., telephone poles). This happens because in the low resolution images small features are not visible and in the high resolution images the disparity search range is not sufficient to match the feature.

An image pyramid is a set of images $I^{(0)}, I^{(1)}, \cdots$ of progressively diminishing resolution that are derived from a common parent image I. Resolution reduction is accomplished by smoothing and the previous layer and then selecting every other pixel. For the data presented in this paper, 4 level pyramids are used, the images are reduced by convolving with a 3×3 Gaussian kernel and selecting every other pixel.

Starting with the lowest resolution images (at the top level), an iterative process is carried out in which a disparity map is computed, expanded to match the size at the next lower level, and refined. This process continues until the final disparity map at the base level is computed. The disparity search range at all levels, except the top level, is $\pm\left(\frac{3p+1}{2p}\right)$ (p is the interval splitting factor described in Section 3). The disparity search range at the top level is set so that the disparity range at the bottom will cover the anticipated range.

At pyramid level k, the initial disparity array $D_0^{(k)}$ is formed by copying the disparities computed at the previous level $D^{(k+1)}$ into every other entry in $D_0^{(k)}$, i.e., $D_0^{(k)}(2i, 2j) = D^{(k+1)}(i, j)$, filling in the missing values in $D_0^{(k)}$ by linear interpolation, and then multiplying the entries in $D_0^{(k)}$ by two. Next, we could simply use $D_0^{(k)}$ to initialize the disparity search at level k, and compute the disparity array $D^{(k)}$ directly by matching $I_R^{(k)}$ and $I_L^{(k)}$ with the search range at pixel (i, j) given by

$$D_0^{(k)}(i,j) - \left(\frac{3p+1}{2p}\right) + \frac{n}{p}, n = 0, \cdots, 3p+1 \quad (4)$$

Or better yet, we could unwarp $I_L^{(k)}$ by making the substitution

$$I_L^{(k)}(i,j) \rightarrow I_L^{(k)}(i + D_0^{(k)}(i,j), j) \quad (5)$$

for all pixels in $I_L^{(k)}$, then compute an incremental disparity array $\Delta D^{(k)}(i, j)$ by matching $I_R^{(k)}$ and $I_L^{(k)}$ (which has just been unwarped) with the search range at pixel (i, j) given by

$$-\left(\frac{3p+1}{2p}\right) + \frac{n}{p}, n = 0, \cdots, 3p+1 \quad (6)$$

and finally update the initial guess to form the disparity array at level k.

$$D^{(k)}(i,j) = D_0^{(k)}(i,j) + \Delta D^{(k)}(i,j). \quad (7)$$

This procedure removes the perspective distortion associated with larger features. Before unwarping $I_R^{(k)}(i, j)$ and $I_L^{(k)}(i + D_0^{(k)}(i, j), j)$ view the same general spot on the surface. Whereas, after unwarping the large scale disparities are removed and $I_R^{(k)}(i, j)$ and $I_L^{(k)}(i, j)$ view the same general spot on the surface. Using $D_0^{(k)}$ to unwarp $I_L^{(k)}$ is similar to the method proposed by Schenk et al. (1980) in which $D_0^{(k)}$ is used to compute an approximate orthonormal image pair from $I_R^{(k)}$ and $I_L^{(k)}$ and then $\Delta D^{(k)}$ is computed by matching the approximate orthonormal images.

5 Terrain reconstruction

The following is a description of the basic steps taken by the terrain reconstruction system (for a detailed description see Schultz (1994)).

1. Resample the raw R and L images into epipolar coordinates.
2. Create n level image pyramids $I_R^{(0)} \cdots I_R^{(n-1)}$ and $I_L^{(0)} \cdots I_L^{(n-1)}$ (see Section 4 for details).
3. Compute the top level disparity map $D^{(n-1)}$ from $I_R^{(n-1)}$ and $I_L^{(n-1)}$ using the weighted cross-correlation match score, subpixel image matching, and hierarchical techniques described in Sections 2, 3, 4.
4. Initialize the level counter $k = n - 2$.
5. Create the initial guess $D_0^{(k)}$ by expanding $D^{(k+1)}$.
6. Unwarp $I_L^{(k)}$.

7. Compute the incremental disparity map $\Delta D^{(k)}$ by matching $I_R^{(k)}$ and $I_L^{(k)}$.

8. Update the disparity map $D^{(k)} = D_0^{(k)} + \Delta D^{(k)}$.

9. Test for the bottom level. If $k > 0$, decrement the level counter $k = k - 1$ and go to step 5, otherwise continue.

10. At the base level, calculate the world coordinate vector $\vec{\mathbf{X}}(i,j) = (X_R^{(0)}(i,j), Y_R^{(0)}(i,j), Z_R^{(0)}(i,j))$ for pixels where $D^{(0)}(i,j)$ exists. This is done by solving the stereo observation equations for all pixel pairs where a correspondence exists [Slama, 1980].

11. Create the orthonormal elevation map $\overline{Z}$ and image $\overline{I}$ by resampling the elevations $Z_R^{(0)}$ and pixel intensities $I_R^{(0)}$ onto a regularly spaced grid in world coordinates.

6 Results

The terrain reconstruction system was tested by processing three sequences of simulated images and one real image pair. To evaluate the performance of system as a function of base-to-height ratio b/h, and with and without the weighted correlation mask and subpixel image matching algorithms, a series of simulated images were analyzed. For each simulation the camera models and locations along with a random surface were specified, and an R and L image pair synthesized using a ray tracing program. Then from the camera models and synthesized images, the surface was recovered and compared to the original simulated one. The same random surface (shown in Figure 1) was used for all simulations. The horizontal dimensions of the surface is $1m \times 1m$, the rms surface height is 1.33cm, and the surface height spectrum is proportional to $k^{-4}cm^{-1}$, where k is the spatial frequency. Furthermore for all simulations, the cameras were located 10m above the surface, the focal length and orientation of the cameras were adjusted so that the entire surface fit within the camera field-of-view, and the optic axis passed through the center of the surface. A series of nine synthesized image pairs were generated for $b/h = (0.25, 0.50, 0.75, 1.00, 1.25, 1.50, 1.75, 2.00, 2.25)$. The simulated image pair for $b/h = 2.25$ is shown in Figure 2. The affects of perspective distortion are clearly visible in this image pair.

The sequence was processed by the terrain reconstruction system described in Section 5 for three sets of parameters, (1) Gaussian weights and $p = 9$, (2) uniform weights and $p = 9$, and (3) uniform weights and $p = 1$.

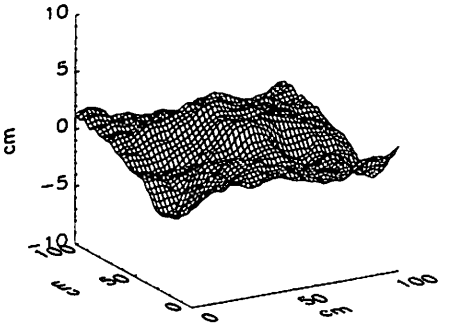

Figure 1: *The random surface used in all simulations.*

Reconstruction errors are reported in terms of the percent of the scene recovered r and the normalized elevation error s. The normalized elevation error s is the standard deviation of the elevation errors for all nodes where an elevation was computed, divided by a normalization factor s_0, i.e.,

$$s = \frac{1}{s_0} STDEV\left(\tilde{Z}(i,j) - \overline{Z}(i,j)\right)$$

where the standard deviation $STDEV$ is computed only for nodes where the recovered elevations $\overline{Z}(i,j)$ exists (see Section 5, Step 11), $\tilde{Z}(i,j)$ are the known elevations, and s_0 is a normalization factor that compensates for the natural improvement of vertical resolution with b/h. The length s_0 is equal to the height of the volume traced out by the intersection of the field-of-view of the pixels at the center of the R and L images [Matthies and Shaffer, 1987].

The simulation results are summarized in Figure 3, where r and s are plotted as a function of b/h for the three sets of parameters described above. Inspection of Figure 3 reveals that when the weighted cross-correlation match score and subpixel image matching algorithms are implemented, r and s do not depend on b/h for values of b/h at least as large as 2.25. If instead, a conventional cross-correlation match score is used, r remains constant and s grow slowly with b/h. If, in addition, integer image shifts are used instead of subpixel shifts, r and s grow more quickly with b/h.

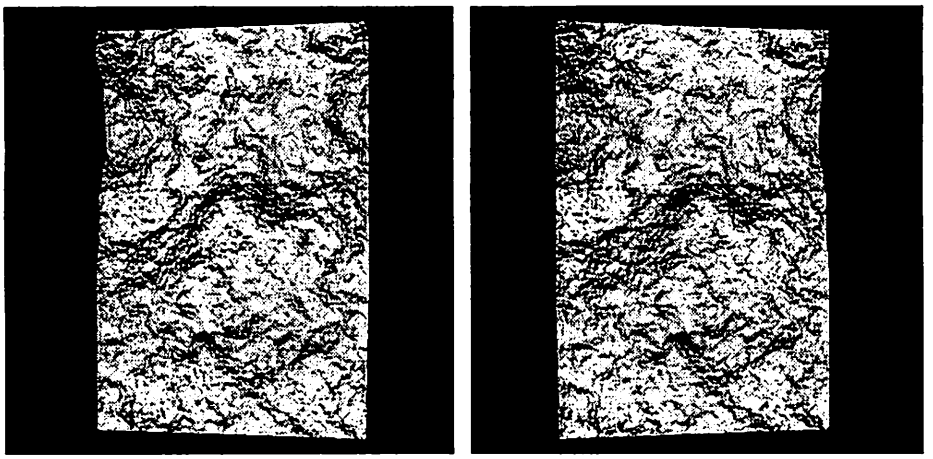

Figure 2: *Synthesized image pair with a 2.25 base-to-height ratio showing a significant amount of perspective distortion.*

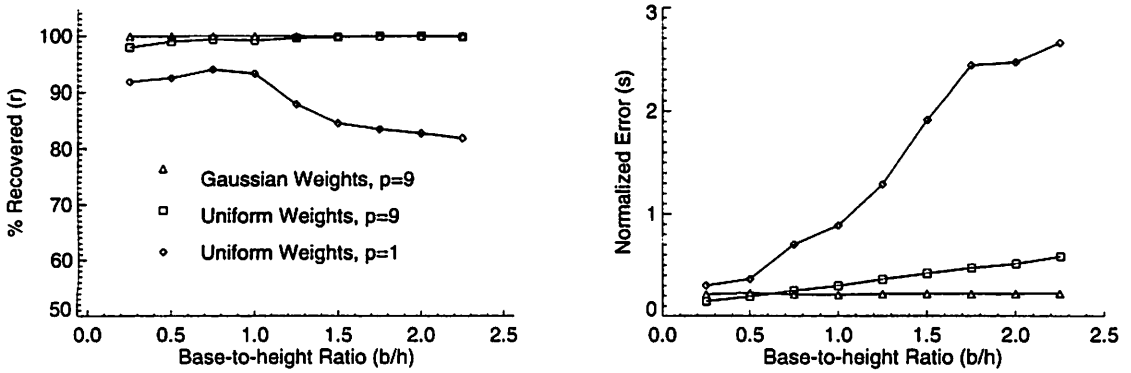

Figure 3: *The percent recovered* (r) *and normalized reconstruction error* (s) *as a function of base-to-height* (b/h) *ratio for Gaussian weights and* $p = 9$*, uniform weights and* $p = 9$*, and uniform weights and* $p = 1$.

In addition to the simulated data, a pair of high altitude photographs shown in Figure 4 of a building, parking lot and surrounding terrain of the Martin Marietta UGV site also were processed. The digitized images along with the camera parameters were supplied by the U. S. Army Topographic Engineering Center. The first of these images was arbitrarily assigned to the R image, while the other one was assigned to the L image. The image pair was then processed using the terrain reconstruction algorithms described above. The images were taken with the cameras looking straight down, with a base-to-height ration of 0.6295. Four level pyramids, Gaussian weights, and subpixel matching with $p = 5$ were used. At the top level the disparity search range was set to $(-12\frac{3}{5}, 13\frac{3}{5})$, and the window sizes for the 4 levels were 5×5, 9×7, 13×11 and 25×21.

The reconstructed orthonormal elevation map $\overline{Z}$ and image $\overline{I}$ are shown in Figure 5. Figures 4 and 5 appear to be rotated and reversed relative to each other because the high altitude images come from digitized negatives and the orthonormal views are displayed in world coordinates. Figure 6 shows rendered views of three areas in the test site—the building, parking lot, and a rock formation. In the rendered view of the building, sharp boundaries, especially corners, are not accurately reconstructed. However, many details of the structure, such as the flat roof and ventilation equipment, are clearly visible. In the rendered view of the parking lot, the basic shape of the cars are visible, however, the light pole is missing (only its shadow remains). This is an expected artifact because the light pole is a small feature with a large disparity. In the rendered view of the rock formation, there does not appear to be any artifacts. This part of the test site has ideal conditions for terrain reconstruction. Notice that the shading on the rocks, vegetation, gully and bare ground

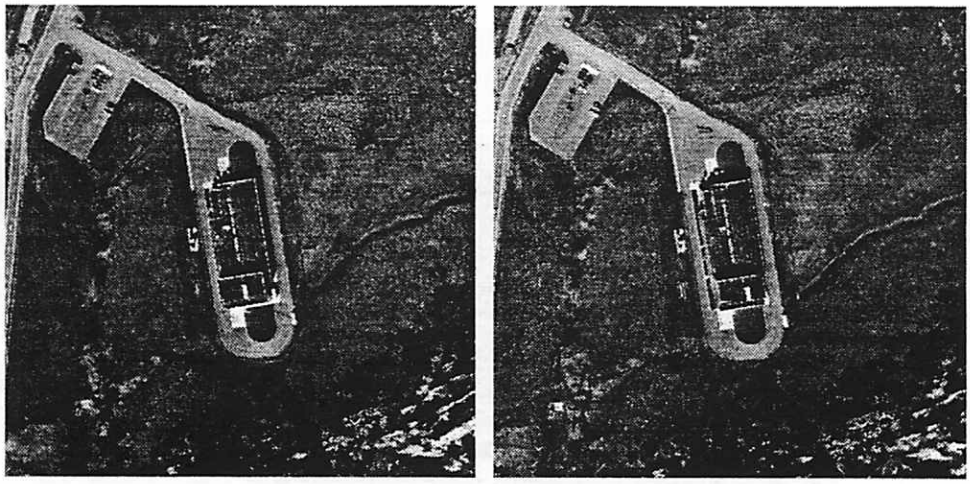

Figure 4: *Two overlapping high altitude photographs of the test site.*

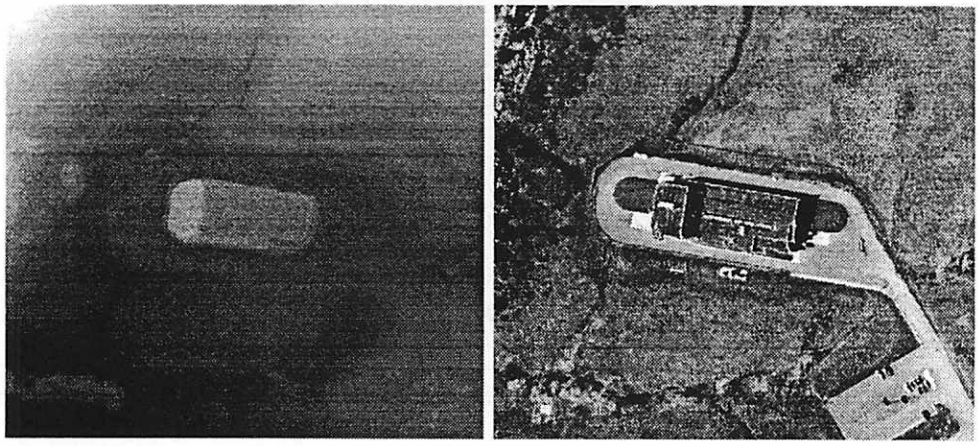

Figure 5: *The elevation map and orthonormal view of the test site.*

are consistent with the shapes of these objects.

7 Conclusions

The algorithms described in this paper were designed specifically to reconstruct terrain from oblique views. Based on analyses of simulated and real data, it appears that terrain can be successfully reconstructed from images taken from widely varying viewpoints. These procedures are especially valuable in operational scenarios, such as stealth navigation and unmanned ground vehicles, where terrain maps must be reconstructed from image data gathered from distant reconnaissance sources. We are currently setting up a series of laboratory experiments to evaluate the performance of the terrain reconstruction system under a variety of operational conditions including b/h, lens focal length, and terrain type. In additions we are in the process of integrating the terrain reconstruction and the UMass automatic building model acquisition systems [Collins et al., 1994, Jaynes et al., 1994].

Acknowledgement

I thank Robert Collins, Bruce Draper, Allen Hanson, Ed Riseman and Richard Weiss for their many helpful discussions and comments during the preparation of this paper.

References

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[2] S. D. Cochran and G. Medioni. 3-d surface description from binocular stereo. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-14(10):981–994, October 1992.

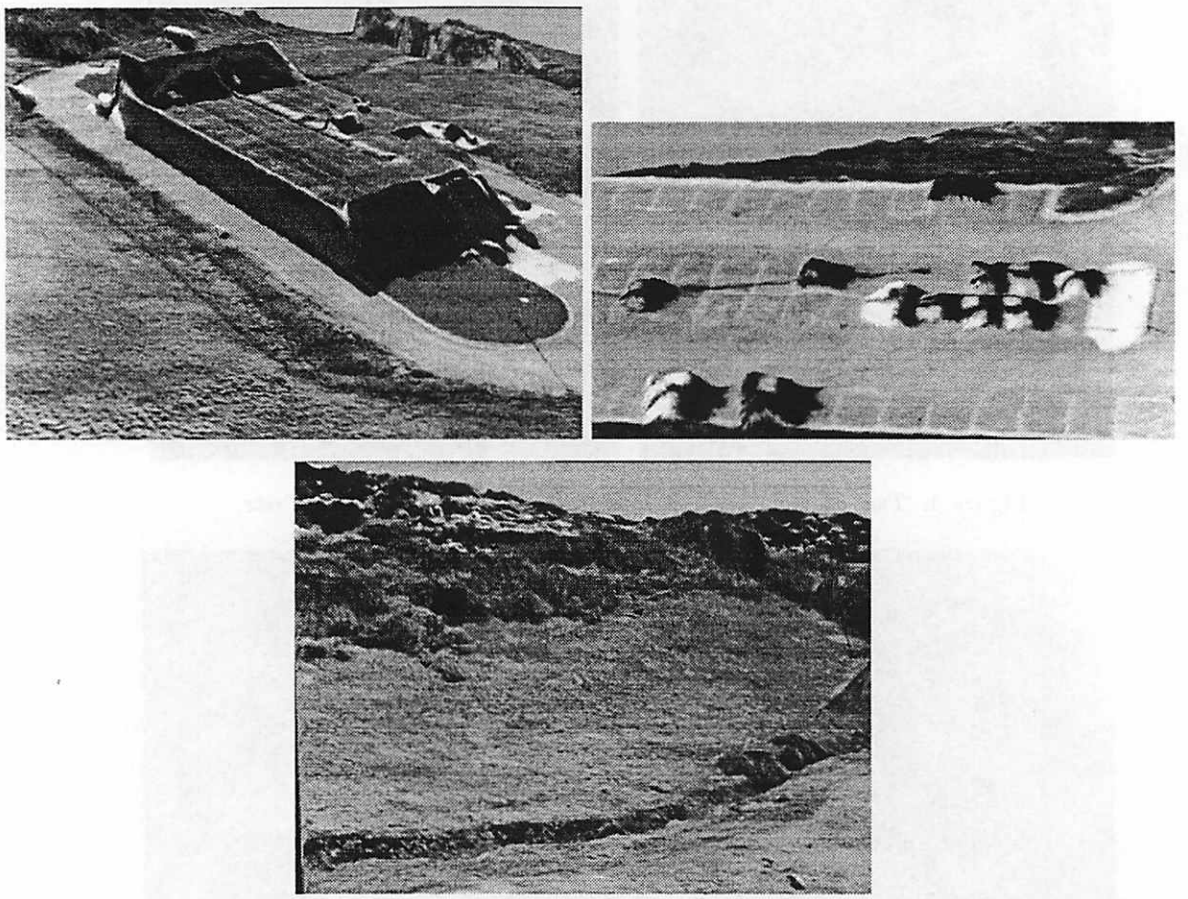

Figure 6: *Rendered views of the test site showing the building, parking lot, and a rock formation.*

[3] R. Collins, A. Hanson, and E. Riseman. Site model acquistion under the umass radius project. In *Arpa Image Understanding Workshop*, Monteray CA, November 1994.

[4] O. Faugeras. *Three-Dimensional Computer Vision, A Geometric Viewpoint.* The MIT Press, Cambridge, MA, 1993.

[5] C. Jaynes, F. Stolle, and R. Collins. Task driven perceptual organization for extraction of rooftop polygons. In *Arpa Image Understanding Workshop*, Monteray CA, November 1994.

[6] L. Matthies and S. A. Shaffer. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):239–248, June 1987.

[7] H. Mostafavi. Image correlation with goemetric distortion. part ii: Effects on local accuracy. *IEEE Transactions on Aerospace and Electronic Systems*, AES-14(3):494–500, May 1978.

[8] T. Schenk, J. Li, and C. K. Toth. Hierarchical approach to reconstruct surfaces by using iteratively rectified imagery. In *Proc. SPIE. Close-Range Photogrammetry Meets Machine Vision*, pages 464–470, Zurich, 1990.

[9] H. Schultz. Terrain reconstruction. Technical Report UM-CS, University of Massachusetts, Department of Computer Science, Amherst, MA, 1994 in preparation.

[10] C. C. Slama, editor. *Manual of Photogrammetry, fourth Edition.* American Society of Photogrammetry, Falls Church, VA, 1980.

[11] Q. Tian and M. N. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics and Image Processing*, 35:220–233, 1986.

Task Driven Perceptual Organization for Extraction of Rooftop Polygons*

Christopher Jaynes, Frank Stolle and Robert Collins

Abstract

A new method for extracting planar polygonal rooftops in monocular aerial imagery is proposed. Through bottom-up and top-down construction of perceptual groups, polygons in a single aerial image can be robustly extracted.

Orthogonal corners and lines are extracted and hierarchically related using perceptual grouping techniques. Top-down feature verification is used so that features, and links between the features, are verified with local information in the image and weighed in a graph structure according to the underlying support for each feature.

Cycles in the graph correspond to possible building rooftop hypotheses. *Virtual features* are hypothesized for the perceptual completion of partial rooftops. Extraction of the "best" grouping of features into a building rooftop hypothesis is posed as a graph search problem. The maximally weighted, independent set of cycles in the graph is extracted as the final set of roof boundaries.

1 Introduction

Extraction of polygonal structures from an aerial image is an important step in building detection and model construction. We would like to determine the shape and location of buildings within an aerial image robustly and accurately by extracting the polygons that define rooftop boundaries.

Industrial and urban centers are typically complex and cluttered with structure. Occlusions, strong perspective effects, and variable lighting conditions are a few of the problems when dealing with aerial imagry of typical urban centers. Despite these difficulties, a successful system will discover rooftops that can be used for further image understanding tasks.

2 Task Driven Organization

The power of perceptual organization for the extraction of structure in natural scenes is well know. [?, 4, 6] In our approach, low level features are perceptually grouped to form *collated features* which are then used to hypothesize the final groupings. However, besides this bottom-up approach, each level of the hierarchy may search for features in a task driven, top-down manner. Grouping choices are driven by the goal of the system and the domain. We apply task driven perceptual organization to the process of polygonal rooftop extraction from aerial imagery.

2.1 Overview

The system proceeds in three steps; low level feature extraction, collated feature detection, and hypothesis arbitration. Each module generates features that are used at during the next phase and interacts with lower level modules through top-down feature extraction.

The low level features in this system are perspective image projections of orthogonal corners and straight line segments in the scene.[1] Mid-level collated features are sequences of corners and lines that are grouped together to form *chains*. High-level polygon hypotheses are formed from closed chains.

Because single collated features can be part of several closed polygons, the final set of closed polygons must be searched for the "best" independent set of closed chains. This is done using certainty measures that are

*This work was funded by the RADIUS project under ARPA/Army contract TEC DACA76-92-C-0041 and also by the National Science Foundation grant No. CDA-8922572

[1] That is, while the corners are orthogonal in the world they are not necessarily orthogonal in the image. The prespective projection is known and the shape of the image corner is computable.

maintained throughout the entire grouping process. As each feature is extracted it is assigned a certainty; the final grouping choice is then found as the independent set of closed chains that maximizes the overall certainty.

2.2 The Feature Relation Graph

Features and their groupings are stored in a graph structure called the *feature relation graph.* Low level features are nodes in the graph, and binary relations between features are represented with an edge between the corresponding nodes. Both nodes and edges are assigned a certainty that reflects the confidence of a feature or a feature grouping.

Cycles in the feature relation graph represent grouped polygon hypotheses. The maximally-weighted, set of independent cycles is then extracted from the feature relation graph to discover a set of independent high confidence rooftop polygons.

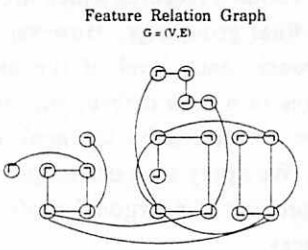

Figure 1: Features are stored in the feature relation graph. Low level features are represented as nodes, collated features as paths, and final polygons as closed cycles.

3 Low Level Feature Extraction

Because we are interested in the detection of human-made structure, orthogonal corners and straight lines were used as low level features. The low level features that are originally extracted are used to form collated features.

3.1 Straight Lines

Straight lines are extracted using two different methods. The primary, bottom-up method for extracting low level straight line features is the Boldt algorithm [1]. This algorithm hierarchically groups edgels into progressively longer line segments based on proximity and collinearity constraints. Figure 2 shows the Boldt lines extracted from a typical aerial urban scene.

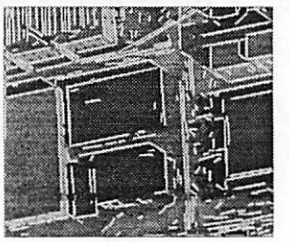

Figure 2: Boldt Lines

Boldt lines are assigned a certainty measure that is calculated during their extraction. The line certainty depends primarily on the contrast of the edge and on the least-squares residual error of the line fit to the grouped edges. For a detailed description of Boldt line certainty, see [1].

Another line detection scheme is used for top-down grouping verification. These *local lines* are extracted when possible groupings between features are being considered. This approach focuses the power of perceptual grouping to a predictive task and avoids reliance on single, globally extracted features.

For example, while attempting to construct a chain feature, it is necessary to discover ad verify if a local line lies between two corners. Each pixel in the image along a connecting line between the two corners is classified as a supporting edgel or nonedgel according to the image intensity gradient, as computed by an oriented Sobel mask, and the variance in the gradient magnitude. This is performed within a rectangular search window between the two corner features.

The final strength of the local line is determined by dividing the number of edgels, L, by the number of pixels in the search line, N. This value is thresholded in order to determine if there is enough edgeness to consider this to be a line. For the results shown here a line threshold of 70% was used.

These local lines are used to verify that a grouping hypothesis, between two corners for example, is justified by evidence in the image. Figure 3 shows a top-down line search between two corner features. The certainty of local lines is based on the contrast of the edge and the percentage of the search window that can be classified as a line.

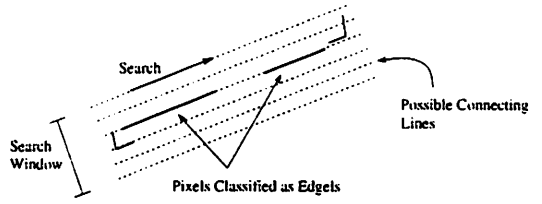

Figure 3: The local line finder is invoked by higher level processes to encourage possible groupings.

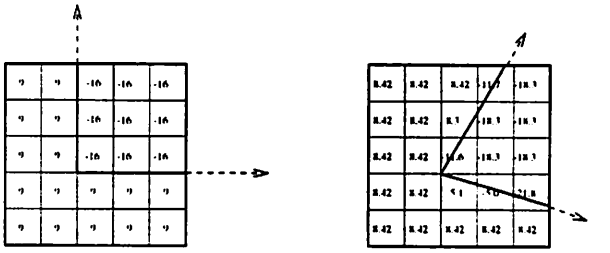

Figure 4: An original 5 × 5 mask and the corresponding perspectively transformed mask that is convolved with the image.

3.2 Orthogonal Corners

Our domain assumption is that rooftop polygons will be produced by flat horizontal surfaces with orthogonal corners. This describes a large majority of the building roofs in urban and industrial centers. Therefore, corner features are orthogonal and parallel to the ground plane in the 3D world. Of course, the apparent shape of an orthogonal corner in the image is not invariant under perspective projection, but varies predictably with respect to image position.

To further simplify processing, we assume that a majority of the buildings are aligned according to an approximate city grid. This assumption reduces the set of orthogonal roof corners to be considered to only four, which for the purposes of this paper are labeled NorthEast, NorthWest, SouthEast and SouthWest. The relative orientation of the city grid with respect to the camera completely determines how these four cardinal corner types will appear in the image. Currently, we compute this orientation from the given camera pose; however the city-grid orientation can also be computed more generally using vanishing point analysis [5]. Once this orientation information is known, the perspective transformation mapping 3D orthogonal corners into 2D image corners can be determined, and ideal corner masks can be generated to accurately extract these important low level corner features.

Four different corner masks are generated. Warping is performed by mapping the lines that define the orthogonal corner through the perspective transformation and into the new expected corner angle. This transformation is performed to sub-pixel accuracy.

The four masks are used to detect each of the possible orientations of a roof corner and to classify the corner's type. Corner types are important for later perceptual grouping of compatible corners.

The final masks, then, are typical $n \times n$ ideal corner detectors that are convolved with the image. In the results shown here 7 × 7 masks were used. Masks of this small size do well in localizing corners and in detecting non-obvious corners (see section 6), however they are sensitive to noise. The performance of template-based corner detection in grey level images with respect to detection, localization, and stability is discussed in [7]. In our system, we allow a large number of false positives when detecting corners and rely on the higher level grouping processes to discard incorrect low level features.

Once constructed, each mask is convolved with the image and the correlation value at each pixel in the image stored. The correlation measure is used as the measure of "cornerness" of each image pixel and is normalized by the maximum change in grey levels in the image over the size of the mask. Normalization is needed because the corners in typical aerial imagery range from high contrast to very dim. The correlation measure at each pixel is the certainty value for the corresponding orthogonal corner feature that is placed into the feature relation graph.

After convolution of each corner mask with the image, a large array of mask responses will be obtained. A large number of false positives are eliminated by thresholding the absolute value of the mask response. For the experiments an empirical threshold of 60% on corner uncertainty was used. Finally, non-maximal suppression over a 7 × 7 window of pixels is used to eliminate neighboring pixels that respond to the corner mask only partially. Figure 5 shows the results of orthogonal corner detection in an aerial scene.

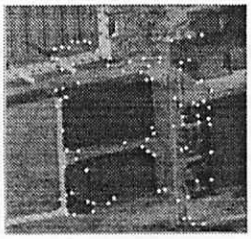

Figure 5: Orthogonal Corners

4 Collated Features

Collated features are constructed from sets of lines and corners extracted from the image. A collated feature is a sequence of perceptually grouped corners and lines that form a chain. A valid chain group must contain an alternation of corners and lines, and can be of any length.

Low level features are grouped together according to the standard perceptual parameters of smoothness and symmetry. When such a group is formed, the corresponding nodes in the feature relation graph are connected with an edge. Paths in the feature relation graph become the chain features.

If a low level feature that is needed to complete a strong perceptual group is missing, a top-down feature detector is invoked and the missing feature is searched for in the image. Currently, the system is able to invoke the local line detector to complete a link of two corners if the lines extracted previously were insufficient.

4.1 Feature Groups

Standard perceptual grouping techniques are applied to the low level features in an attempt to group compatible corners and the line between them. These corner-line-corner triples are the "links" that, when followed as paths in the feature relation graph, form chains. Each link can be thought of as a polygon edge hypothesis, while chains are pieces of a polygon hypothesis. Closed chains are a special case and are treated as completed polygon hypotheses.

In order for a link to be formed, three conditions must be met (Figure ??). Given two orthogonal corners, they must first be of compatible types, where compatibility is defined according to corner type and axis information. For example, the east-pointing axis of a NorthWest corner cannot be grouped with a corner of type SouthWest. It is also not possible for a corner to be grouped with another corner of the same type. Secondly, grouped corners must be in proper spatial alignment with respect to each other. That is, corresponding axes of two corners to be linked must be roughly collinear. Finally, a perceptual link can be formed only

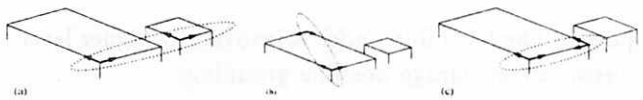

Figure 6: Perceptual grouping of low level features. (a) Incompatible corner types disallow group. (b) Improper alignment of corners. (c) Valid group, supported by line evidence.

if there is evidence for a supporting straight line between the corners. Figure 7 shows an example of perceptually grouped corners.

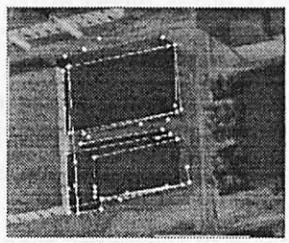

Figure 7: Perceptually Grouped Corner Pairs (Links)

To compute the certainty of a particular chain feature the weight of the corresponding path in the feature relation graph is computed. The certainty of a chain is the sum of the certainties of its parts. Thus, given chain C of length n, the certainty is computed as:

$$\kappa(C) = \sum_{i=0}^{n} \kappa(v_i) + \sum_{i=0}^{n-1} \kappa(e_i) \qquad (1)$$

where v_i is node i in the path corresponding to C, e_i is edge i, and $\kappa(F)$ denotes the certainty of feature F.

5 Polygon Groupings

Extraction of final rooftop polygons proceeds in two steps. First, all possible polygons are computed from the collated features. Then, polygon hypotheses are arbitrated in order to arrive at a final set of non-conflicting, high confidence rooftop polygons.

Polygon hypotheses are simply closed chains, which can be found as cycles in the feature relation graph. All

of the cycles in the feature relation graph are searched for in a depth first manner.

While searching for closed cycles, the collated feature detector may be invoked in order to attempt closure of chains that are missing a particular feature. The system then searches for evidence in the image that such a virtual feature can be hypothesized. Virtual features are hypothesized according to the parameters of perceptual completion. If a cycle is missing a single corner, for example, a virtual corner will be hypothesized at a position that is constrained both by symmetry and smoothness. Currently, the system is able to hypothesize virtual corners and then invoke lower level feature detectors to confirm the hypothesis.

After addition of a virtual corner, the image is searched by the local line finding algorithm for line data that supports this hypothesis. In the event that the evidence is sufficient, the new corner is generated as a low level feature and used to complete the cycle in the feature relation graph.

In this way, high level features do not rely on the original set of features that were extracted from the image. Rather, as evidence for a polygon accumulates, tailor-made searches for lower level features can be performed. This type of top-down inquiry increases the robustness of the system.

Once discovered, all cycles are stored in a dependency graph where nodes represent complete cycles. Nodes in the dependency graph contain the certainty of the cycle that the node represents. An edge between two nodes in the dependency graph is created when cycles have lower level features in common. The final set of polygons then, must be the set of nodes that are both independent (have no edges in common) and of maximum certainty.

A set of polygon hypotheses extracted from a typical image is shown in figure 5. Notice that, with the generation of virtual features such as corners, we are able to complete a polygon that is partly occluded by a neighboring building.

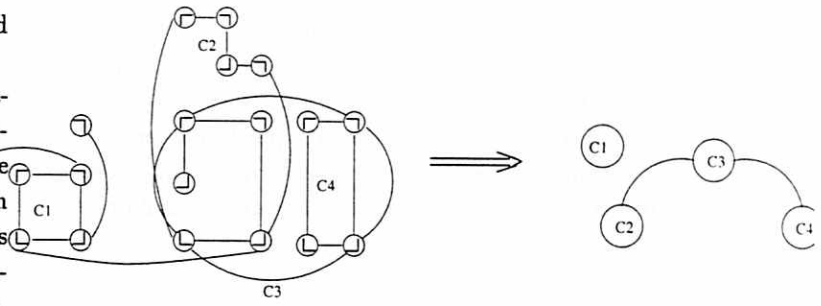

Figure 8: Cycles are extracted from the relation graph and placed, as nodes, into a dependency graph. The maximum independent set of nodes in the dependency graph is the final grouping choice.

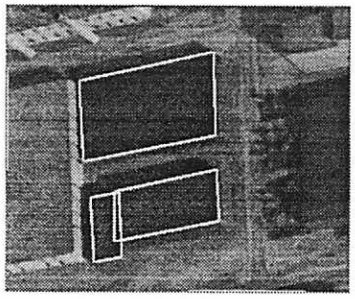

Figure 9: A set of polygon hypothesis extracted from the feature relation graph. When neighboring rooftops partially occlude a polygon, as in the above image, a virtual feature may be generated at the missing corner which can create final polygons that overlap in the 2D projection.

6 Results

In addition to the examples used above, two more examples are shown in order to demonstrate the system's robustness. Both of the images that were used had a variety of buildings, shadows, and many of the difficulties typically found in aerial imagery.

The images used are part of the RADIUS (Research and Development for Image Understanding) model board imagery. As before, the camera model and pose for each image is known. Both images were captured at an approximate height of 10,000 feet above the ground plane. The ground truth data supplied with the model board imagery is used to estimate the accuracy of the system. System accuracy was characterized in several ways:

- Polygons detected versus the true polygons comprising buildings in the image.
- Number of polygon vertices versus number of vertices in building rooftops.
- Average distance between polygon vertices and ground truth rooftop vertices.

Several points, at known rooftop corners in the world were selected for a more detailed performance analysis of the system. Each ground truth point was projected

into the image using the known camera pose. The Euclidean image distance between the ground truth image corner and the extracted polygon corner was computed as the lower bound on the true 3D positional error[2] . This distance gives us an estimate of the polygon placement accuracy with respect to the actual building position and the estimated pose parameters.

6.1 First Test Image

The first example image[10] contains six distinct buildings of varying sizes. The strong shadows and different rooftop heights make this an interesting image for testing purposes. Nine ground truth points were used for an estimation of system accuracy. The low level features extracted are shown in figure 11. The value of virtual feature extraction is shown by building A. A shadow falls across a corner of the building and important low level information is missing. The corners are perceptually grouped and a final set of polygons is generated. The results of the test are shown in figure 12.

Results for Example 1	
Rooftops Detected	100.0 %
Vertex Coverage (No virtual features)	88.5 %
Vertex Coverage (With virtual features)	100.0 %
Avg. Vertex Displacement (Pixels)	3.6

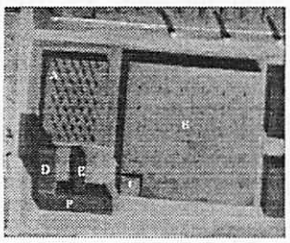

Figure 10: Image used in the first test sequence

6.2 Second Test Image

The second example image contains seven buildings of different sizes and complex shapes. Buildings E and F are very close but are known to be distinct structures. Eight ground truth points were selected for accuracy analysis. As before, the system was run on the image and the performance was analyzed.

[2] At this point, 3 dimensional position of the extracted polygons is not available.

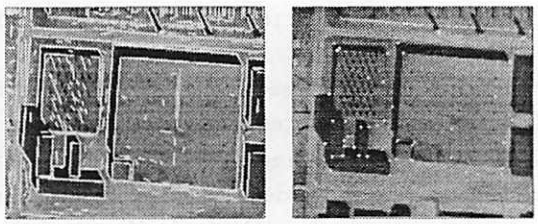

Figure 11: Low level features extracted from the first image: Boldt line data and orthogonal corners.

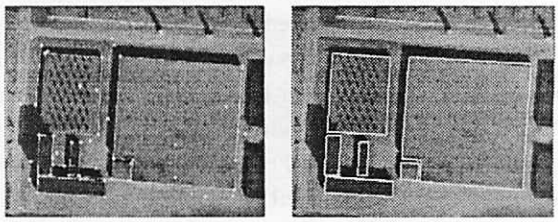

Figure 12: Perceptually grouped corners and the final set of polygon hypotheses

Results for Example 2	
Rooftops Detected	78 %
Vertex Coverage (No virtual features)	86.8 %
Vertex Coverage (With virtual features)	92.1 %
Avg. Vertex Displacement (Pixels)	4.74

Both buildings D and C were not entirely extracted. That is, the final polygons do not match the shape of the underlying structure. Building C is a two level structure and the system failed to discover the lower level rooftop boundary on the right. The corner detector failed to extract crucial low level features at the junction of the two roof heights.

Although the features were extracted on a similar structure to the right of building D, they were not well localized. With small structure, placement error becomes a problem and grouping is difficult. The orthogonal corner detector was designed to extract dihedral corners but can easily be extended to incorporate sun

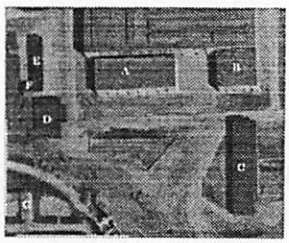

Figure 13: Second test image

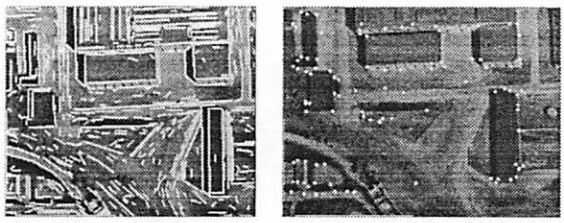

Figure 14: Low level features extracted from the second test image. Boldt line data and orthogonal corners.

angle information for trihedral corner detection. (See Section 7)

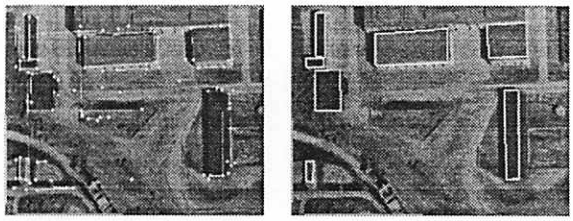

Figure 15: Perceptually grouped corners and the final set of polygon hypothesis

7 Conclusions and Future Work

The results from the proposed approach are encouraging. The system is expected to perform similarly on other aerial images and is currently being tested on a wide variety of aerial photos.

Currently, polygon detection is a piece of the larger aerial image understanding system being developed at U. Mass, Amherst under the RADIUS project. The hypothesized rooftop polygons are verified and refined through multi-image triangulation which computes a height for each polygon in the world. The final set of polygons are extruded to the ground plane for a final volumetric model of buildings.

An improved corner detection mask, that incorporates shadow angles from known sun position, will be constructed. Better methods for solving the maximum independent set problem will be explored, including approximation techniques such as simulated annealing.

The task driven approach to perceptual organization will be expanded to cover more general image understanding tasks. Relaxing restrictions such as flat roofs and orthogonal corners will be investigated so that a more general module can be used to group general structures in aerial imagery.

References

[1] R. Weiss and M. Boldt, "Geometric Grouping Applied to Straight Lines" *IEEE Computer Society on Computer Vision and Pattern Recognition*, 1986.

[2] R. Collins, A. Hanson, E. Riseman, and Y. Cheng. "Model Matching and Extension for Automated 3D Site Modeling." *Proc. DARPA Image Understanding Workshop*, 1992.

[3] M. Herman and T. Kanade. "The 3D MOSAIC Scene Understanding System: Incremental Reconstruction of 3D Scenes from Complex Images" *Proc. DARPA Image Understanding Workshop*, 1984.

[4] A. Huertas, C. Lin, and R. Nevatia. "Detection of Buildings from Monocular Views Using Perceptual Grouping and Shadows" *Proc. DARPA Image Understanding Workshop*, 1993.

[5] J.C. McGlone and J. Shufelt, "Incorperating Vanishing Point Geometry into a Building Extraction System," *Arpa Image Understanding Workshop*, Washington DC, pp.437-448, 1993.

[6] R. Mohan and R. Nevatia, "Using Perceptual Organization to Extract 3D Structures" *Trans. Pattern Analysis and Machine Intelligence*, 1989.

[7] A. Singh and M. Shneier. "Grey Level Corner Detection: A Generalization and a Robust Real Time Implementation" *Computer Vision, Graphics, and Image Processing*, 1990.

[8] V. Venkateswar and R. Chellapa. "Intelligent Interpretation of Aerial Images", *University of Southern California, Dept. of Electrical Engineering Technical Report 137* March 1989.

Triangulation without Correspondences *

Yong-Qing Cheng, Robert T. Collins, Allen R. Hanson, Edward M. Riseman

Department of Computer Science
University of Massachusetts
Amherst, MA. 01003

Abstract

This paper presents two different algorithms for reconstructing $3D$ points from two sets of noisy $2D$ image points without knowing point correspondences given the corresponding poses from the two images. We first present a new way to form a $2D$ similarity function between two points from two images via $3D$ pseudo-intersection. Based on principles of proximity and exclusion, the first algorithm uses a new affinity measure between $2D$ image points from two different images and a competition scheme to establish image point correspondences and recover their corresponding $3D$ points simultaneously. Based on an optimal graph theoretic approach, the second algorithm uses the similarity function to construct a bipartite graph, builds a corresponding flow network, and finally finds a maximum network flow that determines the correspondences between two images. The two proposed algorithms have been applied to aerial images from the ARPA RADIUS project. Experimental results have shown that the proposed algorithms are robust.

1 Introduction

A fundamental and important problem in computer vision is to build $3D$ models of objects and scenes from a sequence of images. So far, extensive research has been done to develop robust algorithms in this area [1-16], including monocular motion sequences, stereo pairs, and a set of distinctive views. The basic principle to deal with this problem is a triangulation process. For a general triangulation process, it is assumed that the intrinsic (lens) parameters and extrinsic (pose) parameters of each camera are known, or that the 3×4 projective transformation matrix which represents a relationship between a $3D$ point and its corresponding $2D$ point is known (as in the RADIUS project). Usually, $2D$ features are extracted first such as corners, curvature points, and lines from each frame of an image sequence. Then, the correspondence of these features is established between any two successive frames, i.e., the correspondence problem. Finally, the $3D$ information is recovered from these $2D$ correspondences in the image sequence. The two most extensively used triangulation algorithms are point-based triangulation and line-based triangulation. The reason to use image lines as an alternative to image points is that lines provide a more stable image feature.

Unfortunately, this basic triangulation process assumes the correspondence problem has been resolved. This has caused criticism and doubts about feature-based methods because the process of finding $2D$ image feature correspondences is time consuming and is difficult to implement reliably. This paper addresses this problem.

In recent years, much work has been done on a variety of correspondence problems [5-16]. Many researchers have worked on the problem of motion estimation without correspondences [5-9,12-16]. Aggarwal et al [8] gave an excellent review of the correspondence problem. Aloimonos, et al. [11], presented an algorithm to estimate $3D$ motion without correspondences by combining motion and stereo matching. Recently, Huang and his research group [7,15-16] presented a series of algorithms to estimate rigid-body motion from $3D$ data without matching point correspondences. Goldgof et al. [7] presented moment-based algo-

*This work was funded by the RADIUS project under DARPA/Army contract number TEC DACA76-92-C-0041 and also by the National Science Foundation grant No. CDA-8922572.

rithms for matching and motion estimation of $3D$ points or lines sets without correspondences and applied these algorithms to object tracking over the image sequences. Lee et al. [9] proposed an algorithm to deal with the correspondence problem in image sequence analysis.

Objects in the world can be nonrigid, and an object's appearance can deform as the viewing geometry changes. Consequently, much work has also been done that addresses the problem of correspondence and description by using deformable models[10-11,17-19]. Scott and Longuet-Higgins [10] developed an algorithm to determine the possible correspondences of $2D$ point features across a pair of images without any other information (in particular, they had no information about the poses of the cameras). They first incorporated a proximity matrix description which describes Gaussian-weighted distances between features (based on inter-element distances) and a competition scheme allowing candidate features to contest for best matches. Then they used the eigenvectors of this matrix to determine correspondences between two sets of feature points. Shapiro and Brady [11] also proposed an eigenvector approach to determining point-feature correspondence based on a modal shape description. Recently, Sclaroff and Pentland [19] described a modal framework for correspondence and description.

In this paper, we first investigate the problem of determining image point correspondences given the poses of two images while simultaneously computing the corresponding $3D$ points. Here, camera pose consists of an orientation R_l and a $3D$ position τ_l which map the world coordinate system to the camera coordinate system. The problem can be formulated as follows:

Given two sets L and R of $2D$ image points from two images I_l and I_r: $L = \{p_i^l(u_i, v_i) \mid p_i^l \in I_l, i = 1, 2, ..., n_l\}$ and $R = \{p_j^r(u_j, v_j) \mid p_j^r \in I_r, j = 1, 2, ..., n_r\}$, and two corresponding poses (R_l, τ_l) and (R_r, τ_r) for the two images I_l and I_r, the goal is to compute a set of $3D$ points $P_q(x_q, y_q, z_q)$ $(q = 1, 2, ..., n,\ n \leq min\{n_l, n_r\})$ representing n correspondences between L and R without knowing in advance the image point correspondences.

First we present a new way to form a $2D$ similarity function between two points from two images via $3D$ pseudo-intersection. Then we present two different algorithms for reconstructing $3D$ points from noisy $2D$ image points without knowing the point correspondences. The first algorithm uses a new version of an affinity measure to extend the work in [10] via the principles of proximity and exclusion. Our second algorithm is based on an optimal graph theoretic approach using the similarity function to construct a bipartite graph, build a corresponding flow network, and finally find a maximum network flow that determines the correspondences between two images. The two proposed algorithms have been applied to aerial images from the ARPA RADIUS project. Experimental results have shown that the proposed algorithms are robust.

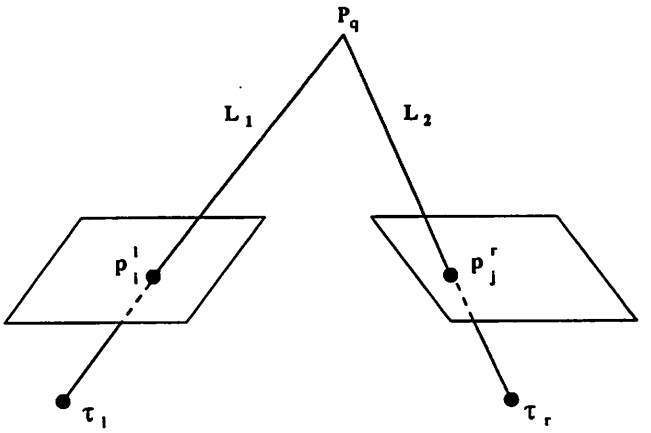

Figure 1: A triangulation process for a pair of images.

2 2D similarity function via 3D pseudo-intersection

Given two poses (R_l, τ_l) and (R_r, τ_r) from two images I_l and I_r, for any pair of $2D$ points p_i^l and p_j^r $(i = 1, 2, \ldots, n_l;\ j = 1, 2, \ldots, n_r)$ from I_l and I_r, there exist two $3D$ lines L_1 and L_2 such that L_1 passes through points p_i^l and τ_l, and L_2 passes through points p_j^r and τ_r, as shown in Figure 1. L_1 and L_2 are the *projection lines* of points p_i^l and p_j^r, respectively.

Suppose each projection line L_k $(k = 1, 2)$ is defined as

$$\frac{x - x_k}{u_{xk}} = \frac{y - y_k}{u_{yk}} = \frac{z - z_k}{u_{zk}} \qquad (1)$$

with unit direction vector $\vec{u_k} = (u_{xk}, u_{yk}, u_{zk})^T$.

Consider first how to compute an optimal $3D$ *pseudo-intersection point* $P_q(x_q, y_q, z_q)$ with the smallest sum of distances from $P_q(x_q, y_q, z_q)$ to two lines L_1 and L_2. The error function can be defined [7] as

$$\begin{aligned} E = \; & [(x_q - x_k)u_{yk} - (y_q - y_k)u_{xk}]^2 \\ & +[(x_q - x_k)u_{zk} - (z_q - z_k)u_{xk}]^2 \\ & +[(y_q - y_k)u_{zk} - (z_q - z_k)u_{yk}]^2 \end{aligned} \quad (2)$$

After setting $\frac{\partial E}{\partial x_q} = \frac{\partial E}{\partial y_q} = \frac{\partial E}{\partial z_q} = 0$, we obtain the optimal $3D$ pseudo-intersection point $P_q(x_q, y_q, z_q)$[7]

$$\begin{bmatrix} x_q \\ y_q \\ z_q \end{bmatrix} = \left[\sum_{k=1}^{2} A_k\right]^{-1} \left[\sum_{k=1}^{2} \left(A_k \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix}\right)\right] \quad (3)$$

where

$$A_k = \begin{bmatrix} u_{yk}^2 + u_{zk}^2 & -u_{xk}u_{yk} & -u_{xk}u_{zk} \\ -u_{xk}u_{yk} & u_{xk}^2 + u_{zk}^2 & -u_{yk}u_{zk} \\ -u_{xk}u_{zk} & -u_{yk}u_{zk} & u_{xk}^2 + u_{yk}^2 \end{bmatrix}$$

If p_i^l and p_j^r are the corresponding image points from two successive images I_l and I_r, then P_q is the real $3D$ point recovered by the traditional triangulation algorithm. However, there are three cases that are exceptions: (1) no $3D$ point could be obtained for p_i^l and p_j^r, because the two $3D$ lines L_1 and L_2 are parallel; (2) an incorrect "negative" $3D$ point could be obtained for p_i^l and p_j^r, due to the two $3D$ lines L_1 and L_2 intersecting behind one or both cameras, as shown in Figure 2; (3) a wrong "epipolar" $3D$ point P_w is obtained corresponding to p_i^l and p_j^r, due to incorrect correspondences, e.g. p_i^l could appear to correspond to either p_j^r or p_w^r as shown in Figure 3. This case shows that a point p_i^l in image I_l could intersect with the projection line of more than one image point in image I_r.

The first case, with parallel projection lines, is exceedingly rare, but is easily be detected by examining whether there exists a solution for Equation (3). It also can be detected by examining whether the directions of the projection lines L_1 and L_2 are the same.

For the second case, as all pairs of image points from two images are considered as possible correspondences, some of those will intersect in their negative directions and satisfy the minimal distance condition to lines L_1 and L_2, but are incorrect. Fortunately, it is easy to detect this kind of

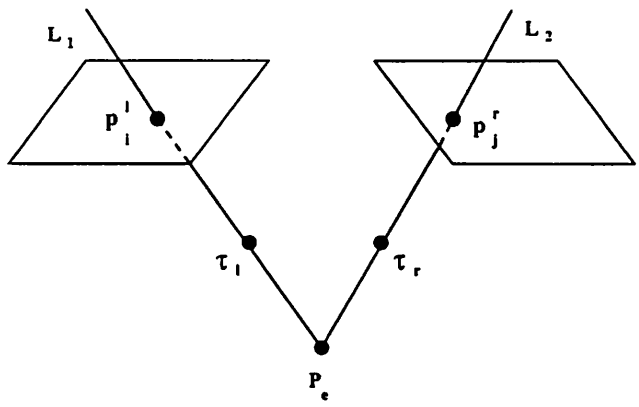

Figure 2: A wrong "negative" 3D point corresponding to a pair of image points.

"negative" $3D$ point by examining the directions of rays $\vec{\tau_l p_i^l}$ and $\vec{\tau_l P_q}$ or $\vec{\tau_r p_j^r}$ and $\vec{\tau_r P_q}$ to make sure that they are the same.

The third case is caused by an incorrect correspondence, often due to ambiguity. For example, as shown in Figure 3, suppose p_i^l corresponds to p_j^r with P_q as the correct $3D$ point. However, the known poses specify epipolar lines, and since p_w^r lies on the known epipolar line of p_i^l in image I_r, then p_i^l and p_w^r could intersect at a $3D$ point P_w. However, this kind of ambiguity might be detected because p_w^l might correspond to point P_w appearing in image I_l. For this case, the maximum correspondences could be detected for two sets of points, i.e., p_i^l corresponds to p_j^r and p_w^l corresponds to p_w^r. Unfortunately, if the point P_w^l doesn't appear in the first image I_l, it is difficult to resolve this inherent ambiguity. In such situations, a third image would greatly reduce such ambiguities.

For any pair of image points (p_i^l, p_j^r), we project the "pseudo-intersection" point P_q into the two images I_l and I_r, then get the two projected image points $p_i^{l'}(u_i', v_i')$ and $p_j^{r'}(u_j', v_j')$:

$$\begin{aligned} u_i' &= S_{xi}\frac{(R_l(P_q)+\tau_l)_x}{(R_l(P_q)+\tau_l)_z} \\ v_i' &= S_{yi}\frac{(R_l(P_q)+\tau_l)_y}{(R_l(P_q)+\tau_l)_z} \\ u_j' &= S_{xj}\frac{(R_r(P_q)+\tau_r)_x}{(R_r(P_q)+\tau_r)_z} \\ v_j' &= S_{yj}\frac{(R_r(P_q)+\tau_r)_y}{(R_r(P_q)+\tau_r)_z} \end{aligned} \quad (4)$$

where S_{ui} and S_{vi} are the intrinsic camera scale factors along the "u" and "v" directions on the image plane I_l, and S_{uj} and S_{vj} are the intrinsic

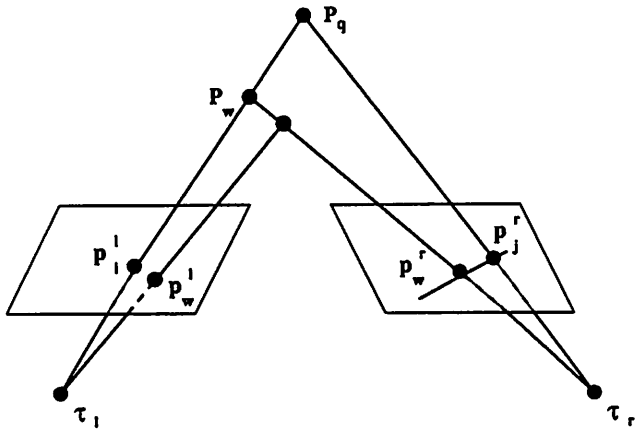

Figure 3: A wrong "epipolar" 3D point corresponding to a pair of image points.

camera scale factors along the "u" and "v" directions on the image plane I_r.

Finally, we compute the error functions E^l_{ij} and E^r_{ij}:

$$E^l_{ij} = \|p^l_i - p^{l'}_i\|_2, E^r_{ij} = \|p^r_j - p^{r'}_j\|_2$$

A $2D$ *similarity function* $sf(p^l_i, p^r_j)$ is defined as

$$sf(p^l_i, p^r_j) = E^l_{ij} + E^r_{ij} = \|p^l_i - p^{l'}_i\|_2 + \|p^r_j - p^{r'}_j\|_2$$

The criterion underlying $sf(p^l_i, p^r_j)$ is that the best estimate for any $3D$ pseudo-intersection point is the point that minimizes the sum of the least-squares distances between the predicted image location of the computed $3D$ point and its actual image locations in the first and second images. if $sf(p^l_i, p^r_j) = \infty$, it means that p^l_i is not similar at all to p^r_j; if $sf(p^l_i, p^r_j) = 0$, it means that p^l_i is perfectly similar to p^r_j.

The next two sections present two algorithms for determining point correspondences using the similarity function $sf(p^l_i, p^r_j)$.

3 Algorithm based on principles of proximity and exclusion

Following the basic idea in Scott and Longuet-Higgins' work[10], we uses pose information to achieve a new powerful version of proximity matrix. The first step is to detect any "negative" $3D$ point P_e and construct a $n_l \times n_r$ proximity matrix H of a Gaussian-weighted error function H_{ij} (i = 1, 2, ..., n_l; j = 1, 2, ..., n_r) using the similarity function

$$H_{ij} = e^{-sf(p^l_i, p^r_j)/2\sigma^2}$$

where σ is the control parameter for the degree of spatial interaction between the two sets of image points.

The second step is to perform a singular value decomposition (SVD) of H, i.e.

$$H = UDV^T$$

where U and V are orthogonal matrices, and D is a diagonal matrix in which the nonnegative singular values appear along its diagonal in descending numerical order.

The final step is to compute the correlation between U's rows and V's columns and obtain an association matrix A:

$$A = UIV^T = UV^T$$

where superscript T denotes the transpose of a matrix. I was obtained by replacing each diagonal element in D by a 1, i.e. I is an identity matrix. Each element A_{ij} indicates the strength of attraction between p^l_i and p^r_j. If A_{ij}=1, there is a perfect correspondence between p^l_i and p^r_j; if A_{ij}=0, there isn't any affinity between p^l_i and p^r_j at all. The affinity between p^l_i and p^r_j is strong only if A_{ij} is largest in both its row and its column.

4 Algorithm based on maximum network flow

The problem of triangulation without correspondences seems on the surface to have little to do with flow networks, but it can in fact be reduced to a maximum-flow problem. In this section, we show how the problem of triangulation without correspondences is formulated as a maximum flow problem on a flow network.

Given the two sets of points L = $\{p^l_i \mid i = 1, 2, ..., n_l\}$ from I_l and $R = \{p^r_j \mid j = 1, 2, ..., n_r\}$ from I_r, then an undirected bipartite graph $G = (V, E)$ can be constructed as follows: $V = L \cup R, E = \{e_{ij}\}$ in which each edge e_{ij} (i = 1, 2, ..., n_l; j = 1, 2, ..., n_r) with a unit weight corresponds to a weighted link between p^l_i in I_l and p^r_j

in I_r if the "distance" between them, defined as $sf(p_i^l, p_j^r)$ is less than threshold T_d, and the corresponding optimal $3D$ pseudo-intersection point P_q computed by equation (3) is not "negative". Here, the threshold T_d is chosen empirically. Obviously, the graph arising in such a case is a bipartite graph by construction, since two points in the same image cannot be linked.

Furthermore, from graph theory, we know that given an undirected graph $G = (V, E)$, a *matching* is a subset of edges $M \subseteq E$ such that for all vertices $v \in V$, at most one edge of M is incident on v. A vertex $v \in V$ is matched by M if some edge in M is incident on v; otherwise, v is unmatched. A *maximum matching* is a matching of maximum cardinality, that is, a matching M such that for any matching M', we have $\mid M \mid \geq \mid M' \mid$. Therefore, the problem of triangulation without correspondences can be considered as the problem of finding a maximum matching in a bipartite graph G.

In order to reduce the problem of a maximum matching in the bipartite graph G to a maximum flow problem in the flow network G', the trick is to construct a flow network in which flows correspond to correspondences. We build a corresponding flow network $G' = (V', E')$ for the bipartite graph G as follows: Let the source s and sink t be new vertices not in V, let $V' = V \cup \{s, t\}$, and let the directed edges of G' be given by

$$E' = \{(s, u) : u \in L\} \cup \{(u, v) : u \in L, v \in R, (u, v) \in E\} \cup \{(v, t) : v \in R\}$$

and finally, assign unit flow capacity to each edge in E'.

The following theorem [21] shows that a matching in G corresponds directly to a flow in the corresponding flow network G'.

Theorem 1 *Let $G = (V, E)$ be a bipartite graph with vertex partition $V = L \cup R$, and let $G' = (V', E')$ be its corresponding flow network. If M is a matching in G, then there is an integer-valued flow f in G' with value $\mid f \mid = \mid M \mid$. Conversely, if f is an integer-valued flow in G', then there is a matching M in G with cardinality $\mid M \mid = \mid f \mid$.*

Intuitively, a maximum matching in a bipartite graph G corresponds to a maximum flow in its corresponding flow network G'. If so, the correspondence problem is equivalent to finding the maximum flow in $G' = (V', E')$, and we can compute a maximum matching in G by finding a maximum flow in G'. It has been shown [21] that if we use the Ford-Fulkerson method, the maximum flow f computed by it can ensure that $\mid f \mid$ is integer-valued, and the cardinality of a maximum matching in a bipartite graph G is the value of a maximum flow in its corresponding flow network G'. Therefore, the correspondence problem can be exactly reduced to finding the maximum flow in G'. Specifically, our algorithm has the following steps:

The first step which is the same as in the first algorithm, is to compute $sf(p_i^l, p_j^r)$ for all possible pairwise matches between any pair of image points (p_i^l, p_j^r). Then, the second step is to generate a bipartite graph $G = (V, E), V = L \cup R$, where L and R are disjoint and all edges in E go between L and R, such that if $sf(p_i^l, p_j^r) \leq T_d$, then there exists an edge in E from p_i^l to p_j^r. The third step is to build the corresponding flow network $G' = (V', E')$ using the above process. The final step is to use the Ford-Fulkerson method to efficiently obtain a maximum matching M from the integer-valued maximum flow. A complete description of their algorithm can be found in [21] and will not be given here.

Since any matching in a bipartite graph G has cardinality at most $min(\mid L \mid, \mid R \mid) = O(\mid V \mid)$, the value of the maximum flow in G' is $O(\mid V \mid) = O(n_l + n_r)$. We can therefore find a maximum matching M in a bipartite graph G in time $O(\mid VE \mid)$. For each vertex, the number of edges which is incident on the vertex can be considered as a constant. Thus, the time complexity is approximately $O(\mid V \mid^2) = O((n_l + n_r)^2)$.

5 Experimental Results

In this section, we will illustrate the two algorithms and demonstrate their performance on images $J1$ and $J2$ (shown in Figure 4) from the RADIUS "Model Board 1" set.

In order to demonstrate the robustness of the two algorithms, they were compared in the presence of noise. In general, there are two sources of error which contribute to the error of 3D points recovered from two images: (1) localization errors of the 2D image points and (2) errors in the estimate of the intrinsic and extrinsic camera parameters.

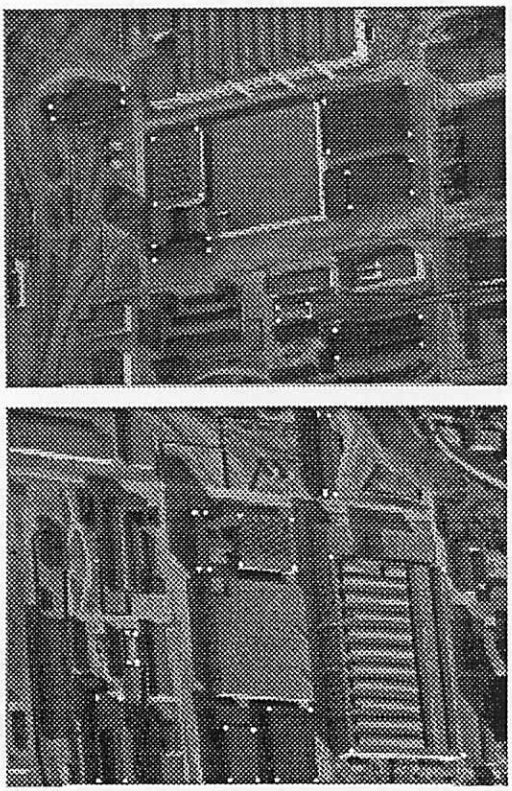

Figure 4: A pair of RADIUS images.

The contribution of the error of the 2D image points from two images is complicated. To aid the error analysis in determining the correspondences between the two images shown in Figure 4, 32 ground truth corner points were selected and projected into each image as shown by the white dots. The image point locations from each image were corrupted by Gaussian noise. Noise for each image point location was assumed to be zero-mean, identically distributed, and independent. The standard deviation ranges from 0.5 pixel to 4.0 pixels. For each level of noise, 100 noisy sample sets were created and the two algorithms were run on each of the samples. From each sample run, the number of incorrect correspondences and the squared distance error between the triangulated point position and its ground truth position were computed. The average number of incorrect correspondences and average triangulation error for each noise level are shown in Figures 5-6, respectively. The experiments have shown that the two algorithms work very well if there is no difference in sizes of image points from two images, i.e. for two images, each 3D point to be recovered

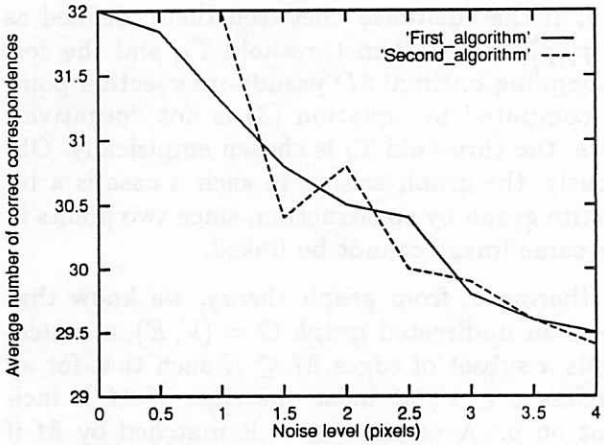

Figure 5: Comparison of number of correct correspondences for the two algorithms as a function of noise.

appears in each image and only their correspondences are unknown.

In order to show how the number of incorrect correspondences is affected by the number of missing points (i.e. no correct correspondence in the other image), new data sets were created by randomly deleting some percentage of image points from the same two sets of 32 image points used above. Also, four levels of Gaussian noise from 1 pixel to 4 pixels were added to these new data sets. The average number of incorrect correspondences over 100 sets of samples for each noise level were computed. Figures 7 and 8 show how the performance of the two algorithms was affected by the number of missing points and the noise. Our experiments have shown that the two algorithms can tolerate a difference in the number of image points from two images and are robust against a reasonable level of noise.

The experimental results [20] also show that it is very difficult to choose an appropriate value for the parameter σ in the first algorithm and a σ that is too large can not be chosen. For the first algorithm to work well, a sufficiently large σ must be chosen, yet it must not be too large since thsi would drive the smaller singular values towards zero. If this occurs, the association matrix becomes unstable. This conforms the conclusion reached in [11].

The second algorithm utilizes a threshold T_d on the error function to build the initial bipartite graph. This threshold must be chosen empiri-

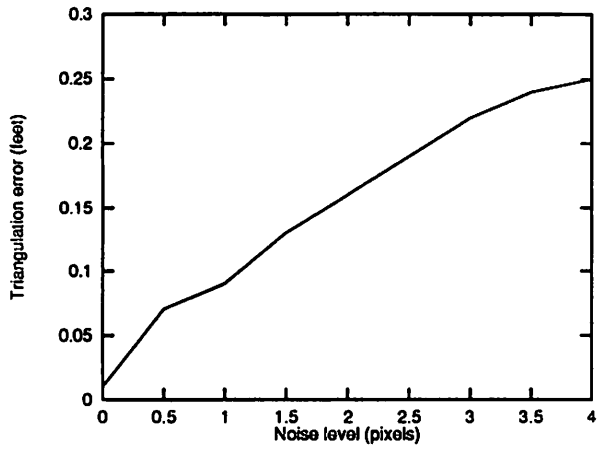

Figure 6: Average triangulation error for the two algorithms.

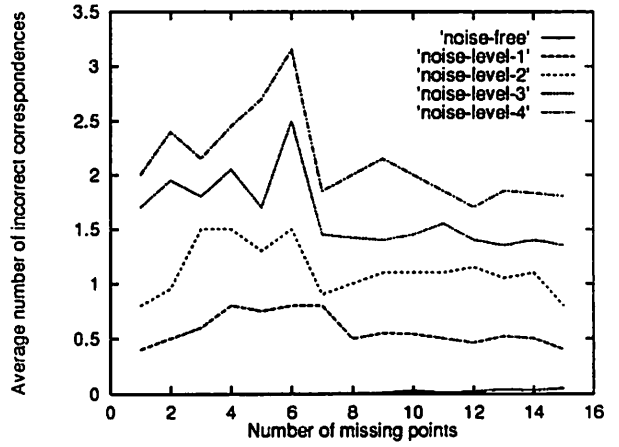

Figure 7: Comparison of number of incorrect correspondences for the first algorithm.

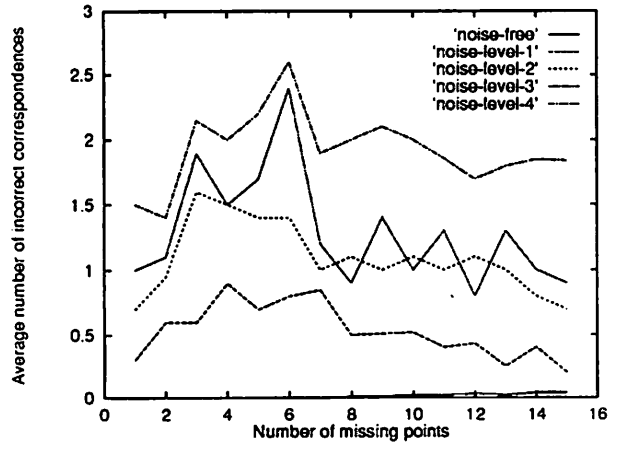

Figure 8: Comparison of number of incorrect correspondences for the second algorithm.

cally. Our experiments showed that it is crucial to choose an approriate value for the threshold T_d.

6 Conclusions and Future Work

Based on a similarity function between two points from two images via $3D$ pseudo-intersection, this paper present two different algorithms to reconstruct 3D points from noisy 2D image points without knowing the point correspondences. The first algorithm is based on a principle of proximity and a principle of exclusion. It first builds a proximity matrix to represent the affinities, then does a SVD decomposition of the proximity matrix to get an association matrix, and finally obtains the correspondences from the association matrix. The second algorithm first reduces the problem of triangulation without correspondences to that of a maximum matching problem of the bipartite graph, then reduces the maximum matching problem to a maximum flow problem of the flow network, and finally determines the correspondences by finding a maximum network flow from the flow network.

The work presented in this paper compared the two algorithms, in terms of robustness with respect to noise and the difference between the set sizes of image points from two images. The experiments showed that the two algorithms are robust. The two algorithms do have several advantages: (1) they can automatically detect the outliers from the $2D$ image points from a pair of images by thresholding the error function for left and right pseudo-intersection projections; (2) they are not too sensitive to noise or the difference between the set sizes of image points from two images; (3) their computational complexity is low; (4) they can be expanded to reconstruct $3D$ lines from noisy $2D$ image lines. From the preliminary experiments conducted thus far, it appears that there is only a small difference between the two algorithms relative to the difference in the size of the point sets. However, there are several reasons to choose the network flow algorithm over the first one: (1) potential instabilities in the association matrix (as described earlier); (2) computational consideration in the association matrix; (3) the network flow is easily extended to multiple images.

For the two algorithms, however, there is some inherent ambiguity which cannot be distinguished.

For such cases, additional images are needed. Currently, reconstruction from $2D$ image points and lines without correspondences is a component of the image understanding system being developed at our computer vision lab under the RADIUS project.

In the future, we are examining ways of modifying the second algorithm so that the threshold T_d is not needed. This will involve introducing mechanisms for weighting in the matching problem and the maximum flow formulation. Furthermore, this algorithm will be extended to perform triangulation from multiple images without knowing the correspondences.

References

[1] Steven D. Blostein and Thomas S. Huang, "Quantization errors in stereo triangulation", Proc. of First International Conf. on Computer Vsion, pp. 325-334, 1987.

[2] Z. Zhang and O. D. Faugeras, "Tracking and grouping 3D line segemnts", Proc. of 3rd International Conf. on Computer Vision, pp. 577-580, Osaka, Japan, 1990.

[3] R. Kumar, "Model dependent inference of 3D information from a sequence of 2D images", Ph.D dissertation, Dept. of Computer Science, University of Massachusetts, Amherst, 1992.

[4] R. Deriche, R. Vaillant and O. D. Faugeras, "From noisy edge points to 3D reconstruction of a scene: a robust approach and its uncertainty analysis", Proc. of the Second Europeian Conf. on Computer Vision, pp. 71-79, 1992.

[5] T.-C. Chou and K. Kanatani, "Recovering 3D rigid motions without correspondences, Proc. ICCV, pp. 534-538, June 1987.

[6] R. Szeliski, "Estimating motion from sparse range data without correspondences", Second Int. Conf. on Computer Vision, pp. 207-216, December 1988.

[7] D. B. Goldgof, H. Lee and T. S. Huang, "Matching and motion estimation of three-dimensional point and line sets using eigenstructure without correspondences", Pattern Recognition, Vol.25, No. 3, pp. 271-286, 1992.

[8] J. K. Aggarwal, L. S. Davis and W. N. Martin, "Correspondence processes in dynamic scene analysis", Proc. of IEEE 69, pp. 562-572, 1981.

[9] Chia-Hoang Lee and Anupam Joshi, "Correspondence problem in image sequence analysis", Pattern Recognition, Vol. 26, No. 1, pp. 47-61, 1993.

[10] G. L. Scott and H. C. Longuet-Higgins, "An algorithm for associating the features of two patterns", Proc. Roy Soc Lond, Vol. B244, pp. 21-26, 1991.

[11] L. S. Shapiro and J. M. Brady, "Feature-based correspondence: an eigenvector approach", Image and Vision Computing, Vol. 10, No. 5, pp. 283-288, June, 1992.

[12] J. Aloimonos and I. Rigoutsos, "Determining 3-D motion of a rigid surface patch without correspondences under perspective projection", Proc. of AAAI, pp. 681-688, August 1986.

[13] A. Basu and J. Aloimonos, "A robust algorithm for determining the translation of a rigidly moving surface without correspondence for robotics applications", Proc. of IJCAI, pp. 815-818, August 1987.

[14] E. Ito and J. Aloimonos, "Is correspondence necessary for the perception pf structure from motion?", Image Understanding Workshop, pp. 921-929, April 1988.

[15] H. Lee, Z. Lin and T. S. Huang, "Finding 3-D point correspondences in motion estimation", Proc. of Eighth Int. Conf. on Pattern Recognition, pp. 303-305, 1986.

[16] H. Lee, Z. Lin and T. S. Huang, "Estimating rigid-body motion from three-dimentional data without matching point correspondences", Int. J. Imaging Systems Technol. 2, pp. 55-62, 1990.

[17] A. Pentland and S. Sclaroff, "Closed-form solutions for physically-based shape modeling and recognition", IEEE PAMI, 13(7), pp. 715-729, July 1991.

[18] A. Pentland and B. Horowitz, "Recovery of non-rigid motion and structure", IEEE PAMI, 13(7), pp. 730-742, July 1991.

[19] S. Sclaroff and A. Pentland, "A modal framework for correspondence and description", Proc. of IEEE, pp. 308-313, 1993.

[20] Y. Q. Cheng, R. Collins, A. Hanson, and E. Riseman, "Triangulation without correspondences (I)", Technical Report (to appear).

[21] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, "Introduction to algorithms", The MIT Press, 1990.