3D Model Acquisition using a Bayesian Controller*

Christopher O. Jaynes, Mauricio Marengoni, Allen Hanson, and Edward Riseman

Computer Vision Laboratory
Dept. of Computer Science
Box 34610, University of Massachusetts
Amherst, MA. 01003
E-mail: jaynes@cs.umass.edu, URL: http://vis-www.cs.umass.edu

Abstract

The problem of automatic model acquisition from computer images has a long and sophisticated research history. Recently, there has been a focus on the interpretation of aerial imagery for model reconstruction and there has been significant progress in several sub-areas. However, a general model acquisition system that is not restricted to a small class of models and contexts has not been realized.

This paper presents a framework that separates control decisions and knowledge from the underlying image understanding (IU) modules. The control system is able to make use of IU modules by applying them in the correct context and interpreting their results according to the current state of knowledge. A set of Bayesian networks are used to represent knowledge about objects and object relationships, and to allow the control system to select an action that is expected to decrease the uncertainty about the object under consideration.

The framework is applied to the problem of automatic building reconstruction from aerial images. Results show how the framework extends the capability of the Ascender system, a building model acquisition system, by classifying image regions prior to geometric reconstruction.

1 Introduction

The extraction and reconstruction of geometric models from images is an important focus of the computer vision community. Significant progress has been made in several constrained subareas and systems perform reasonably well within the domains for which they were designed. These (sub)efforts can be characterized by the contextual restrictions embedded into the algorithms at the time they were designed. These implicit and explicit restrictions vary from the type and characteristics of the data required for processing to the classes of objects addressed by the algorithm. Although these algorithms perform well within the particular contexts for which they were designed, they often degrade significantly within different domains.

A more general solution to the problem lies in the systems ability to automatically select the appropriate algorithm, apply it to the correct subset of the data, and interpret the result according to the current context and knowledge. The framework presented here allows for the integration of these special purpose algorithms into a large knowledge-based system.

This idea of a (set of) local expert(s) that are applied within the correct context, and then fused together into a coherent interpretation of the scene is not new. Object specific-experts, referred to as schemas, played a prominent role in early work on the Schema system [Draper'89], several knowledge directed vision systems [Rimey'92, Musman'93, Sarkar'95], as well as other reconstruction systems from the aerial image domain [Chellapa et al.'94, Huertas and Nevatia'80, Gifford and McKeown'94, Jaynes'96a, Matsuyama'85]. Under this model, robustness is achieved by providing multiple reconstruction/recognition strategies which are applicable under well defined conditions and generality is achieved by increasing the number of object classes to describe a larger fraction of the world.

The framework is founded on three fundamental principles: 1) Specific image understanding strategies are clearly successful under particular contexts for a particular class of objects but may break down when applied in contexts that exceed the design constraints. 2) Domain knowledge, knowledge acquired from partial processing of the data, and knowledge about available image understanding strategies are all valuable in constraining the reconstruction problem. 3) A successful system will contain many specific strategies but will selectively apply them in the correct context, with the correct set of parameters, and will fuse the results of individual strategies into a complete reconstruction.

In order to demonstrate these principles, we apply the

*Funded by the APGD project under DARPA contract number DACA76-97-K-0005, Army Research Office under contract number DAAG55-97-1-0188, and in part by the National Council for Scientific Research-Brazil grant number 260185/92.2.

framework to the problem of automatic model acquisition from aerial images. The framework is applicable in any domain where a set of specialized visual processes are available to gather evidence about the scene. The utility of knowledge-directed processing is demonstrated by a detailed comparison of an existing site reconstruction system and the same system, augmented with our knowledge-based framework. The Ascender system [Collins, et al.'96] has been shown to effectively reconstruction a single class of buildings from a set of aerial image. We show that the functionality and accuracy of Ascender and similar systems can be extended through the explicit use of knowledge.

Ascender detects polygons in a single image through the grouping of image lines into chains based on geometric constraints. All possible groupings are searched for polygon chains that may represent a building boundary. A set of building polygons are then matched across multiple views of the site in order to compute building heights. We introduce the knowledge-directed reconstruction framework to the Ascender system in order to improve its generality, robustness, and overall performance. Image polygons are classified using the network and reconstructed according to the classification. The new framework removes false positives, allows for a more efficient reconstruction through the selective application of available algorithms, and incorporates both domain knowledge and context that can be re-engineered according to different domains without having to recode the underlying algorithms.

The overall knowledge-directed framework is discussed in the next section followed by a more detailed description of the two main system components; the visual subsystem and the knowledge base. A discussion of how the framework is used to extend the capability of Ascender and systems like it can be found in section 3. Finally, we compare the results of the knowledge-directed system with Ascender and discuss how knowledge has improved the automated extraction of a site model.

2 System Overview

The system is divided into a visual subsystem and a knowledge base. The *visual subsystem* contains a library of IU algorithms, a geometric database that contains available data (images, line segments, functional classifications, etc.), as well as models that may have been acquired through processing. Display of the acquired models and a user interface is supplied by the Radius Common Development Environment (RCDE) [Mundy et al.'92], a geometric modeling package. The *knowledge base* is based on belief networks and is constructed using HUGIN [Andersen'89], a system for designing belief networks and influence diagrams. The knowledge base consists of reasoning mechanisms, a control system, and a set of belief networks that represent knowledge about the site. The two systems communicate through Unix socket IP mechanisms. Figure 1 shows an overview of the system.

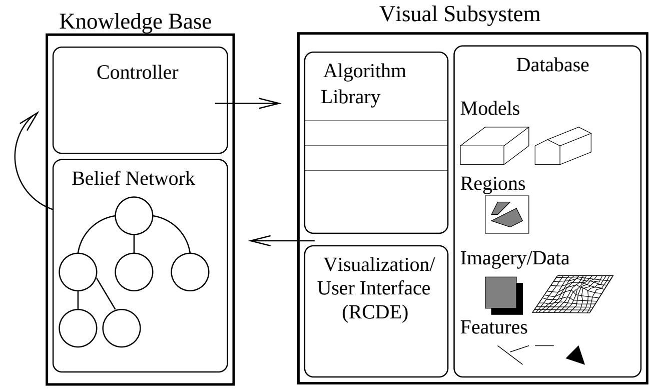

Figure 1: System overview. Control decisions are based on the current knowledge about the site. Vision algorithms, stored in the visual subsystem, gather evidence about the site, update the knowledge base and produce geometric models.

Reasoning takes place over *regions of discourse* that represent a subset of the available data. Regions of discourse may be image regions, a particular building model, or other sets of data that may have been produced by the system. As opposed to systems that partition the space under consideration *a priori* [Rimey'92], regions of discourse are formed, merged, and destroyed during processing of the data. For each region of discourse that the controller selects, a processes is forked to begin the processing.

2.1 Hierarchical Bayesian Knowledge-Based Control

A Bayesian network is a probabilistic inference system that uses a graph representation to denote causal dependencies within the domain under consideration. Each node in the network represents a random variable and each arc represents a relationship between the variables. More details about Bayesian networks can be found in Jensen [Jensen'96].

Bayesian network systems has been used as an inference mechanisms in several domains. In particular, they have been used in computer vision applications such as, control of vision processes for symbolic interpretation of a scene [Rimey'92], classification complex objects, such as a ships, based on shape [Musman'93] or recognition of object structures based on image features [Jian-Ming'93], information fusion and control in a multi agent system [Jensen'92] and management of computational resources[Sarkar'95].

The Bayesian network controller for Ascender II is a system that is used for classification of polygons of aerial images. The controller is a hierarchical system composed of a set of networks divided by different levels of detail. Each level provides relevant information about the objects at a particular scale of detail. Processing within the network is restricted to a single level until classification at the root node occurs. This classification is used to begin processing within a new network that is related to classification at the correct level of detail. Given enough processing resources and time, the process continues until the root nodes of at least one of the networks have been classified. This is presented in Figure 2

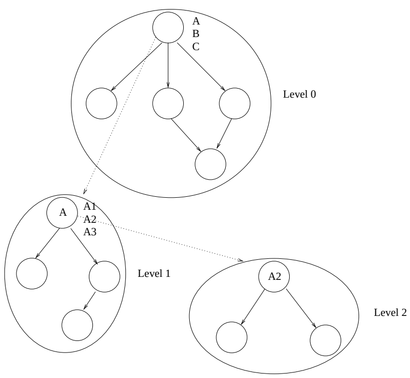

Figure 2: The controller starts in level 0 and finds some outcome for the root node at that level (A, B or C). If the outcome is A and time for further computation is available the controller loads the network for A in level 1 (the doted line shows this inference call). The process can be repeated to subsequent level until the finest level is reached or there is no more time for computation.

Each network in this system represents structural knowledge about features that are expected to be within the domain. For example, a height node may involve information about the expected height of buildings as opposed to that of a grassy field. Specific site knowledge can influence processing of the scene and is stored in the prior probability distributions at each node in the network. That is, if 80% of the regions detected in a certain area are buildings and the only two possible classifications for a region are either building and open field, then the expected frequency of a particular region being an open field is 20%, and the expectation of finding regions heights greater than zero will be proportionally higher than regions of small height.

The vision operators are selected based on the uncertainty of the random variables in the network. Consider the case where plan fit and line count are both boolean variables, if a belief for a good plan fit in a region is 54% and a belief for high number of lines inside the region is 67%, the uncertainty is higher for the plan fit variable, so this operation will be selected in order to reduce the uncertainty. An action is selected to reduce the uncertainty in the node using the expression:

$$node = arg\ min_n(max(Belief(n)) - \frac{1}{S_n})$$

Where S_n represents the number of states of node n. Once a node is selected a request is passed to the visual subsystem for the application of a particular algorithm to a selected region. The findings of this action are then returned to the controller, entered as evidence and propagated through the network.

It is well known that the propagation of evidence in Bayes nets is, in general, an NP-hard problem [Cooper'90] and the time for propagation is a function of number of nodes, number of links, structure of the network and number of states per node. Instead of using a large network and propagate evidence through branches that will not affect the overall classification process, the system uses small networks, which will increase performance, and propagates evidence locally.

A certain operator can be called in different levels, but because each call is related to a certain region, if a call is made twice for the same region the system will not compute the new value, instead it will return the value computed previously and stored in a geometric database.

The decision process on a certain state in the root node is done relatively to the other states, that is after each evidence is propagated through the network all states in the root node are verified, the maximum belief and the second maximum are computed and compared. If the maximum is at least twice the value of the second maximum the controller stops and gives to the region the label defined by the state with the maximum belief. This belief has a minimum bound which is given by:

$$min\ Bel = 2 * \frac{1}{(S_r + 1)}$$

Where S_r represents the number of states in the root node.

Although there is an overhead for controlling the networks in different levels and checking states in root nodes after each evidence, this system is expected to perform better then a ruled based system, since it will not make an exhaustive call for all possible algorithms in the vision module, an better then a system using a single network, mainly because it will improve the performance for evidence propagation.

2.2 Visual Subsystem

The visual subsystem is comprised of two parts; a function library that stores the set of IU algorithms available to the system, and a geometric database that contains available data in the form of imagery, partial models, and other collateral information about the scene (such as classification of functional areas). Each subset of the data, either given prior to processing as knowledge, or acquired through processing, is referred to as a *region of discourse* or simply region. For example, regions may be three dimensional volumes in the scene (ie- building model), 2D image areas (ie-focus of attention area), or even disparate collections of image features (ie-all pixels with a particular grey scale value).

At the request of the controller, an algorithm is selected from the library and run on a region that currently resides within the geometric database. New regions may be produced as a result of processing and these are stored in the database for future reference. For example, initial processing of a set of data begins with a large region of discourse containing many different classes of objects. The system invokes a process to find image polygons that may represent building rooftops. Each of these regions may lead to new regions representing different classes of objects.

The algorithm library contains information about each of the algorithms available to the system for selection and application to the data as well as a definition of the contexts in which each algorithm can be applied. For example, an algorithm that computes a planar fit error to a particular region can only be run in the context of an available range image. Either the range image is already available (in the form of an Interferometric Synthetic Aperture Radar image, for example) or must be able to be computed (from two overlapping views of the region and an available, stereo-optical algorithm, for example). The context information for each algorithm must be explicitly defined when the algorithm is placed into the visual subsystem library.

The library of algorithms presented here were developed to address aspects of the site reconstruction problem from aerial images. For example, finding regions that may contain buildings, classifying building rooftop shapes, and determining the position of other cultural features, are all important tasks for the model acquisition system.

Many of the IU algorithms may be very "lightweight", are expected to perform only in a constrained top-down manner, and may even be used in more than one context. This is due to the fact that the IU algorithms are responsible for gathering evidence for a particular hypothesis put forward by the controller. For example, an algorithm that detects the presence of local maximum in a region of the elevation data can be viewed as a car detector when invoked on a parking lot area. The same algorithm may detect the presence of a rooftop structure when applied to a known building area.

Algorithms may also be very sophisticated, such as the reconstruction of flat roof buildings from multiple views (the role of the Ascender system). Below, the algorithms used to extend the Ascender system are described.

Line Count The line count algorithm extracts and counts line segments from a given region in an optical image. Line segments are detected using the Boldt line algorithm [Weiss'86]. This algorithm hierarchically groups edgels into progressively longer line segments based on proximity and collinearity constraints. Extracted line segments are associated with the region to which they belong and are stored in the Geometric Database for future use. Straight lines provide important clues to feature classes and can be used to find higher level image features such as junctions and polygons. Figure 3 shows a set of line segments extracted from an aerial image.

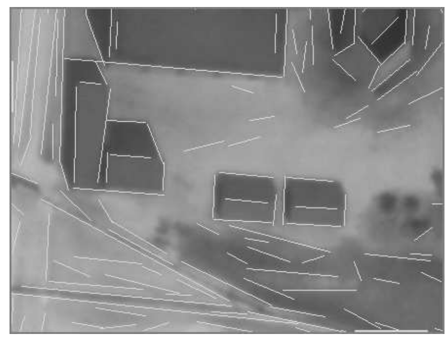

Figure 3: Extracted line segments for a region within an optical image. Line segments may assist in discriminating object classes. In this example, the existence of a centerline may be due to a peaked roof building.

Junction Count This algorithm computes the number of junctions within a particular region of discourse. Junctions are computed by intersecting lines that lay within the region. For each intersection the line segments are back-projected to a nominal Z-plane in the world to compute an angle of intersection. The controller specifies the type of junction to count and, given the relative angle and position of the line segments the junction is either counted or not. For example, both L and T junctions must be near orthogonal in the world and the relative position of the two lines determines if the intersection is type L, T, or neither. Figure 4 shows both L and T junctions detected in an image region.

Average Junction Contrast Similar to the junction count routine, the algorithm computes the line intersections within the region of a particular type. The contrast for a single intersection is the average of the grey-scale contrasts across the two line segments that gave rise to the intersection. The algorithm returns the average contrast of all valid intersections within the region.

Planar Surface Fit If more than one view of a region is available, elevation data can be computed through a stereo-optical routine. The elevation data can then be used as input to robust surface fitting techniques. In this

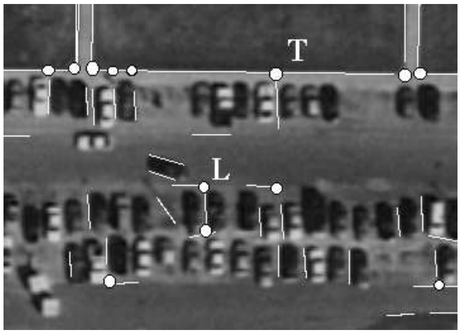

Figure 4: Line junctions provide evidence that a cultural feature may exist within an image region and lend evidence to a region classification.

algorithm, a planar surface is fit thorough the region and a percentage, related to the residual fit error, is returned to the knowledge base. The percentage represents how well a planar model can be fit through the region's elevation data. Elevation data provides a rich source of evidence about a region.

Region Height The region height is computed through a multi-image matching scheme. The matching-scheme used in Ascender I is used here.

Region Size Ratio A ratio of the smallest region dimension versus the largest is returned. For rectangular regions this is a ratio of the smallest side versus the longest.

Edge Terminals Search a region boundary for areas that may have a line termination that is not part of a region corner. These "breaks" in line segments may have been grouped together when the region was produced but may still provide evidence about the region class. Figure 5 shows a region of discourse and the several new regions produced by a search for the regions edge terminals. The new regions can then be searched for evidence that may lead to a classification of the parent region.

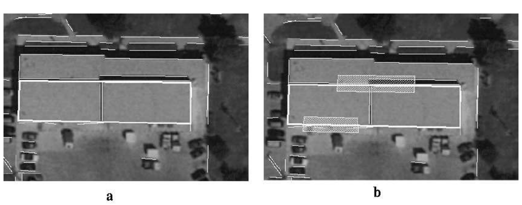

Figure 5: A search along an existing region (a) for line terminals that are not explained by the region corners produces new regions of discourse (b). These regions can be search for evidence that a multi-level building is present (as in the figure).

In all, the type and number of algorithms must be sufficient to distinguish between the different object classes and assist in the model acquisition process. Algorithms are associated with nodes in the belief network if they are capable of gathering evidence relevant to the belief that the node represents. For example, a poor plane fit within a region may imply that the region is not a flat roof building, therefore, the algorithm **Planar Fit** is associated with the node in the network that corresponds to building rooftop class.

The algorithms listed above were chosen because they are relevant to domain in which the system will be tested. However, if the framework is to be truly general useful, the cost of adding a new algorithm to the system must not be prohibitive, something that proved to be a problem in earlier knowledge-based vision systems [Draper'89]. Only two components are necessary to convert an IU algorithm into an evidence policy that are usable by the system.

First, the context in which the algorithm is intended to be run must be defined. Currently, the definition of allowable contexts is straightforward and only disallows algorithms to be run in invalid contexts (on the wrong type of data, for example). This is similar to the Context Sets introduced in the Condor system [Strat'93]. This definition of context is expected to be too simple for our needs and eventually the framework will be extended to allow the definition of a performance profile for each algorithm that defines the expected performance of the algorithm under a variety of different contexts. Secondly, a method for deriving a certainty value from the output of the algorithm must be defined. This certainty value is used by the system to update the knowledge base using Bayesian inference.

3 Experimental Results, Extending the Ascender System

An experiment was conducted to demonstrate how the introduction of knowledge directed framework into the site reconstruction process can improve the accuracy of the final site model. The dataset contained seven overlapping aerial views of the site. The area contains building, parking lots, road networks, and many other cultural features typically found at a urban site. Elevation data for the site was precomputed using a stereo-optical system.

The experiment was performed in two stages. First, a hand-crafted building reconstruction system was executed on the area under consideration[Collins, et al.'96]. The system was developed over several years and tuned to extract flat rectilinear buildings. The system detects rectilinear structure in a single image of the dataset and uses the known relative camera pose between other views to compute a polygon height. This final site model is a set of these boundaries extruded to the ground. Although the system has been shown to be effective in detecting a large percentage of buildings at the site, production of false positives in the site model can be a problem. For example, during the 2D polygon detection phase, a parking lot region is segmented due to accidental alignment of several cars and sufficient line evidence within the image (figure 6a). Although the Ascender system attempts to eliminate these false positives by searching other images for sufficient edge evidence, often this is not sufficient to discriminate between true buildings and false positives. These regions are then produced as final models within the site (figure 6b).

A

a **b**

Figure 6: Typical errors produced by the Ascender reconstruction system. (a) Accidental alignment in a single image produces a false rooftop hypothesis. (b) Multi-image matching finds sufficient support for a complete 3D model at an incorrect height.

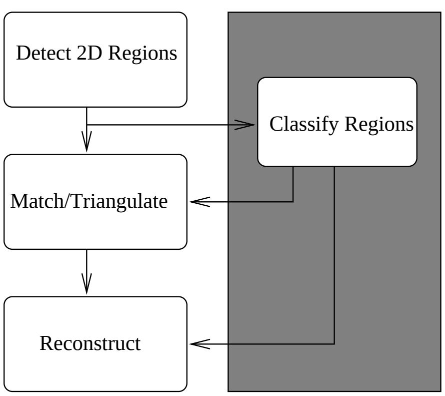

Figure 7: Experimental flow of control. Left column shows the hand designed reconstruction system. After 2D polygons are generated, the knowledge base attempts to classify each region. Only buildings are matched and triangulated for a 3D model while other regions are reconstructed separately as labeled "functional areas".

Because the we are interested primarily in the classification of potential building areas, the knowledge base system had only 2 networks. The first level classifies a region into one of the classes (`Building, Parking Lot, Open Field, Complex, Other`).

A second network attempts a finer classification of building regions into either (`Multi-level Building, Single Level Building`). The first level network is shown in figure 8 and the building class-specific network is shown in figure 9.

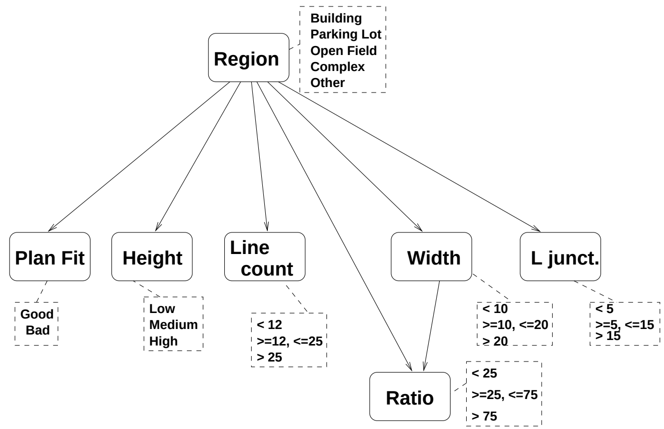

Figure 8: This network works in the coarsest level in tries to classify a region into one the possible outcomes: Building, Parking Lot, Open Field, Complex, Other

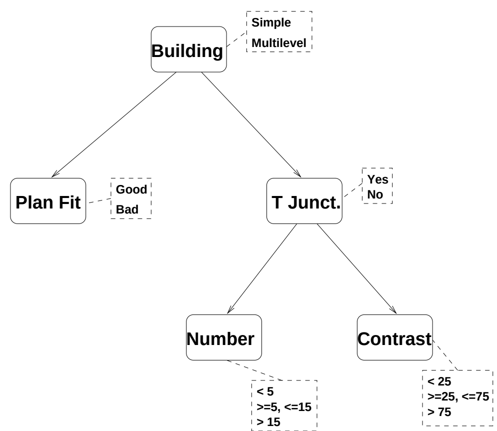

Figure 9: This network works in level 2. For each building found in level 1 this network is invoked and tries to classify the building as either a simple building or a multilevel building.

For each polygon, a region was produced by the visual subsystem and a request for classification was issued to the knowledge base. After the selection of an appropriate evidence policy, the action selected is passed to the visual system where the actual processing is accomplished. Evidence values are returned to the knowledge base where they are used to update the network. The system was run on the 42 regions shown in figure 10a, the distribution of the regions is presented in the table below, and it stopped when a belief value for one of the states reached the limit condition or the controller was unable to select a new action. The region is then classified according to the state of `footprint class` with the maximum belief value found so far. The result classification using the controller is presented in figure 10b.

Region Type	Total
Simple Buildings	21
Multilevel Buildings	1
Parking Lots	4
Complex	1
Open Fields	13
Unknown	2

In figure 10 region "C", which is a parking lot, was classified as a building, partly due to the corresponding elevation data which was uncharacteristically smooth for most parking lot regions. Region "D", which is composed of a parking lot and building with some area of grass should have been classified as complex, but the system classified it as parking lot. This mistake is understandable in light of the the fact that the `complex` classification includes a mixture of features from all other models. Parking lot features, such as many short lines and a rough elevation map, are not only present in region "D" but prevalent. The other two regions that were misclassified are the two small regions in the parking lots, a car at right of region C, that was classified as open field instead of unknown and a truck at left of region "A" which was also classified as open field. Region "B" was classified correctly as a building in level 1 but in level 2 the system exhausted all actions and was not able to decide between simple building and multilevel building. The maximum belief obtained for that region was 66% for simple against 34% for multilevel. Region "A", which is a parking lot, had the same problem in level 1, the system exhausted all actions and at the end the highest belief presented was for parking lot (59%) and the second highest value was for open field (31%). The only multilevel building in the scene, region "E" was properly classified.

In the overall classification process the system used only about 41% of the actions available. An interesting result in this process is that an area of the same type, say "open field", was classified using a different set of actions. The system always started with "plan fit", because of the prior probabilities for each outcome, and in some cases, depending on the outcome of "plan fit", the sequence used was just ratio and height but sometimes the system asked also for other actions such as width before deciding on open field.

Another interesting issue in the controller is that not all nodes in the knowledge base are boolean variables. For instance, the plane fit node is set with a likelihood value based on the result from the visual module. The final models achieved with the use of the knowledge framework and without are compared in the figures below.

4 Future Directions

We have demonstrated how a flexible, knowledge-directed control framework can improve the scope and accuracy of model acquisition systems such as Ascender. Our goal is to demonstrate that this flexibility improves system performance and widens its scope of applicability. To this end, work is underway on engineering the software architecture of Ascender II and on the development of additional evidence policies for a wider range of building classes. The general framework being employed supports any type of data as long as there are corresponding evidence policies available for interpreting it. Consequently, the system is being extended to include IFSAR elevation maps (in addition to elevation maps from traditional stereo techniques) and multi-spectral imagery for improved ground classifications. We expect to use the Fort Hood image dataset as well as other datasets as they become available (e.g. Ft. Benning) to demonstrate the Ascender II system.

There are many issues to be addressed during the design and implementation of Ascender II. One issue concerns the granularity of the IU algorithms employed in the system and how this affects system performance. For example, should Ascender I be dismantled into component parts and reassembled in the knowledge network? Previous attempts to build knowledge-based systems ran into major knowledge engineering problems. The treatment of IU algorithms as black-box evidence gathering mechanisms, regardless of the underlying complexity, may be one way to avoid this. Currently, simple greedy evidence policy is being used to select the next action. What other policies are reasonable and how do the affect the system efficiency? Techniques that compare the expected utility of applying a particular evidence policy to its expected cost will be investigated as one way to answer the question of efficient control.

References

[Andersen'89] S. Andersen, K. Olesen, F. V. Jensen,F. Jensen, "HUGIN - A shell for building Bayesian belief universes for expert systems" In *Proceedings of the 11th international joint conference on artificial intelligence*, pp 1080-1085, 1989.

[Weiss'86] R. Weiss and M. Boldt, "Geometric Grouping Applied to Straight Lines," *IEEE Computer Society on Computer Vision and Pattern Recognition*, 1986.

[Breese'95] J.S. Breese and D. Heckerman. "Decision-theoretic case-based reasoning", Microsoft Research, MSR-TR-95-03, January 1995.

Figure 10: (a) Regions produced by the Ascender system used for classification. 22 polygons correspond to building rooftops. (b) Classification results. Four regions were misclassified and in two other regions the system was not able to decide on the classification. Letters referred to in the text.

[Chellapa et al.'94] R. Chellapa, L. Davis, C. Lin, T. Moore, C. Rodriguez, A. Rosenfeld, X. Zhang, and Q. Zheng. "Site-Model-Based Monitoring of Aerial Images" *Computer Vision and Pattern Recognition (CVPR)*, pp. 694-699, 1997.

[Collins'95] R.Collins, Y.Cheng, C.Jaynes, F.Stolle, X.Wang, A.anson and E.Riseman, "Site Model Acquisition and Extension from Aerial Images," *International Conference on Computer Vision,* Cambridge, MA, June 1995, pp. 888–893.

[Collins, et al.'96] R. Collins, C. Jaynes, Y. Cheng, X. Wang, F. Stolle, A. Hanson, E. Riseman. "The ASCENDER System: Automated Site Modelling from Multiple Aerial Images", Submitted to: Special Issue in *Computer Vision and Image Understanding (CVIU)* on Building Detection and Reconstruction from Aerial Images, guest editors R. Nevatia, A. Gruen, to appear 1998.

[Cooper'90] F. Cooper, "The Computational Complexity of Probabilistic Inference using Bayesian Belief Networks", *Artifical Intelligence*, vol. 40 pp. 393-405. 1990.

[Draper'89] B. Draper, R. Collins,J. Brolio, A. Hanson, E. Riseman. "The Schema System", *International Journal of Computer Vision*, vol. 2. pp. 209-250. 1989.

[Draper'96] B. Draper, A. Hanson, E. Riseman, Knowledge-Directed Vision: Control, Learning, and Integration. pp. 1625-1637.

[Gifford and McKeown'94] J. Gifford, D. McKeown. "Automating the Construction of Large-Scale Virtual Worlds *Proc. ARPA Image Understanding Workshop*, 1994.

[Herman'94] M. Herman and T. Kanade. "3D Mosaic Scene Understanding System: Incremental Reconstruction of 3D Scenes from Complex Images". *Proc. ARPA Image Understanding Workshop*, 1994.

[Horovitz'90] . Horovitz, "Reasoning about Beliefs and Actions Under Compuational Resourcce Constraints", *Proc. of 3rd Conference on Uncertainty in Artifical Intelligence*, pp 301-342, 1989.

[Huertas and Nevatia'80] A. Huertas and R. Nevatia. "Detecting Buildings in Aerial Images" *Computer Vision, Graphics, Image Processing.* vol. 13, 1980.

[Jaynes'94] 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.

[Jaynes'97] C. Jaynes, E. Riseman, and A. Hanson, "Building Reconstruction from Optical and Range Images" *Computer Vision and Pattern Recognition (CVPR)*, San Juan Puerto Rico, June 1997.

[Jaynes'96b] C. Jaynes, F. Stolle, H. Schultz, R. Collins, A. Hanson, and E. Riseman. "Three-Dimensional Grouping and Information Fusion for Site Modeling from Aerial Images" *Proc. ARPA Image Understanding Workshop*, pp. 479-490, 1996.

[Jaynes'96a] C. Jaynes, R. Collins, Y. Wang, F. Stolle, H. Schultz, A. Hanson, and E. Riseman. "Automatic Construction of Three-Dimensional Models of Buildings", chapter to appear in

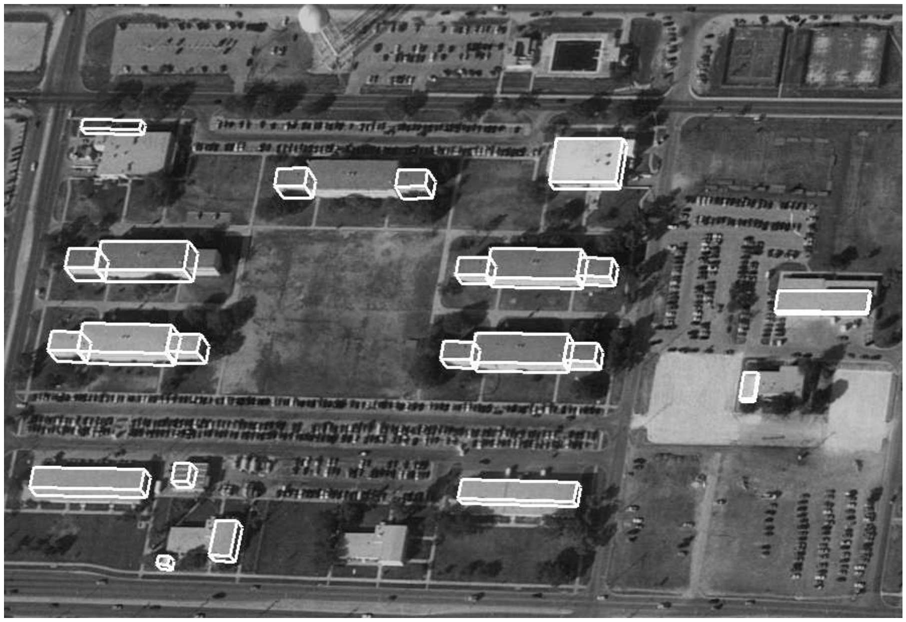

Figure 11: Site model acquired using the knowledge-directed (Ascender II) framework. Regions classified other than building can remain in the database for future use but do not clutter the site model.

ARPA IU RADIUS book. Oscar Firschein and Thomas Strat (eds.), Morgan Kaufmann Publishers, San Francisco, CA, 1997.

[Jensen'92] F. Jensen, H. Christensen, and J. Nielsen, "Bayesian Methods for Interpretation and Control in Multi-Agent Vision Systems", *Applications of Aritifical Intelligence X: Machine Vision and Robotics, SPIE Proceedings Series*, Vol. 1708, 1992.

[Jensen'96] F. Jensen, *An Introduction to Bayesian Networks* Springer Verlag New York, 1996.

[Jian-Ming'93] L. Jian-Ming, I. Christensen, and F. Jensen, "Qualitative recognition using Bayesian Reasoning", Technical Report Aalborg University - Institute of Electroonic Systems, November 1993.

[Lin et al.'94] C. Lin, A. Huertas, R. Nevatia "Detection of Buildings Using Perceptual Grouping and Shadows", *Computer Vision and Pattern Recognition (CVPR)*, pp. 62-69, 1997.

[Liu'95] C. Liu and M. Wellman, "On State-Space Abstraction for Anytime Evaluation of Bayesian Networks" *SIGART Bulletin*, vol. 7(2) pp. 50-57, 1995.

[Matsuyama'85] T. Matsuyama and V. Hwang, "SIGMA: A Framework for Image Understanding: Integration of Bottom-Up and Top-Do wn Processes," *Proceedings of the Ninth IJCAI*, Los Angeles, CA, pp. 908-915, 1985.

[Mundy et al.'92] J. Mundy, R. Welty, L. Quam, T. Strat, B. Bremner, M. Horwedel, D. Hackett, and A. Hoods, "The RADIUS Common Development Environment", *DARPA Image Understanding Workshop*, pp. 215-226, 1992.

[Musman'93] S. Musman and L. Chang, "A study of Scaling Issues in Bayesian Belief Networks for Ship Classification", *Proc. of 9th Conference on Uncertainty in Artifical Intelligence*, pp 32-39, 1993.

[Rimey'92] R. Rimey and C. Brown, "Task-Oriented Vision with Multiple Bayes Nets", Active Vision, eds. A. Blake and A. Yuille, MIT Press, 1992.

[Russel'95] S. Russel and P. Norvig. Artificial Intelligence, A Modern Approach, Prentice Hall: 1995.

[Sarkar'95] S. Sarkar and K. Boyer, "Using Perceptual Inferece Networks to Manage Vision Processes" *Computer Vision and Image Understanding*, vol. 62(1) pp. 27-46, 1995.

[Strat'93] T. Strat. "Employing Contextual Information in Computer Vision", *Proc. ARPA Image Understanding Workshop*, 1993.

[Zilberstein'96] S. Zilberstien, "Using Anytime Algorithms in Intelligent Systems", *AI Magazine* vol. 17(3), pp. 73-83, 1996.