

Detecting Changes in Aerial Views of Man-Made Structures

Andres Huertas and Ramakant Nevatia*

Institute for Robotics and Intelligent Systems

University of Southern California

Los Angeles, California 90089-0273

{nevatia|huertas}@iris.usc.edu

www: iris.usc.edu

Abstract

An important application of machine vision is to provide a means to monitor a scene over a period of time and report significant changes. Comparing intensity values of successive images is not effective as such changes don't necessarily reflect actual changes at a site but might be caused by changes in the view point and illumination. We also want to ignore seasonal changes and focus on structural changes instead. We take the approach of comparing a 3-D model of the site, prepared from previous images, with new images to infer significant changes. This task is difficult as the images and the models have very different levels of abstract representations. Our approach consists of several steps: model to image registration aims to bring the site model and the images into close correspondence; model validation to confirm the presence of model objects in the image; structural change detection seeks to resolve matching problems and provide cues indicating possibly changed structures; model updating aims to suggest modified models for existing model structures and to incorporate new models. Our system is able to detect changes, such as missing (or mis-modeled) buildings, changes in model dimensions, and new buildings under some conditions. Several experimental results are presented.

Keywords: Structural change detection, image to model correspondence, model validation, building detection, aerial image analysis, vision application

* This research was supported by Contract No. DACA-76-93-C-0014 from the Defense Advanced Research Projects Agency (DARPA) and monitored by the Topographic Engineering Research Center of the U.S. Army.

1 Introduction

One of the key applications for aerial and space image analysis is in monitoring and keeping track of changes on the ground. This could be for various purposes such as urban planning, agricultural analysis, environmental monitoring and military intelligence. Currently, such analysis is conducted by human *image analysts*, however, this is a tedious and time consuming task for the humans and allows for only a small fraction of the available imagery to be examined. Many kinds of changes could be relevant such as in structures on the ground, in natural vegetation and in environmental conditions. We focus on changes in man-made structures such as construction or modifications of roads or buildings. Changes in the desired structures should be reflected in some changes in the image, not all changes in the images may be caused by 3-D structural changes. Image intensities can vary due to a number of factors such as changes in illumination, viewpoint, and atmospheric conditions. Further, seasonal variations may cause changes in vegetation and ground cover; while these represent “real” changes on ground, they may not correspond to structural changes.

There is little previous work in structural change detection. Early work in change detection was based on determining pixel intensity changes [1], which do not necessarily correspond to the desired changes as explained above. Structural change detection has similarities to the problems of industrial inspection where a new image is compared with a reference image (or a model) to detect significant variations (see for example, [2]); however, in industrial settings, viewpoint and illumination conditions are typically fixed and controlled. In outdoor scenes, we must deal with the environmental conditions as they are and may have only limited control over how the data gets collected. Thus, to find structural changes, we should not compare images taken at different times directly but rather compare new images (or descriptions derived from them) to an abstract model derived from previous observations; such models have come to be called *site models* [3]. This approach also allows an updating of the site models which may be one of the prime goals of the change detection analysis.

Detecting changes at a site based on abstract models from previous observations poses several significant challenges for computer vision techniques. A new image can not be directly corresponded to an abstract model. Instead, we must compute descriptions from the image that can be corresponded with the model or descriptions that can be derived from the model. This is a common problem in object detection in computer vision and various techniques such as *alignment* have been developed to solve it [4]. The change detection problem is simpler to the extent that some parameters of the *pose* of the objects may be known *a priori*. However, the objects themselves may have changed and not fit a prior model exactly. Also, aerial images typically contain a large number of man-made and natural objects, not all of which may have been modeled (for example, we do not assume models for trees in a scene). The objects of interests may be partially (or totally) occluded by other objects and shadows cast by them may cause confusion. The images also contain significant amount of texture, thus leading to a large number of features at the lower-levels (such as edges) that prohibit use of combinatorial techniques to search for desired objects. Finally, we need to verify that the suspected changes actually correspond to some 3-D structures and to derive a description of the changes where possible.

The system described in this paper is designed for detecting changes in 3-D building structures, though many of the techniques may also apply to other objects such as roads. We do not handle mobile objects, such changes are addressed, for example in [5]. We further assume that the build-

ings are rectilinear and that composite shapes (such as “L” or “T”) are described by their rectangular components, thus allowing us to have a uniform representation for the rectilinear shapes. Each rectangular part is represented by its 3-D wire-frame (consisting of 8 vertices and 12 edges). Our techniques should generalize to other shapes but some of the detailed procedures would have to be changed. We assume that the camera geometry and approximate viewpoint from which images are taken are known. Specifically, we assume that the errors are such that a projected model can be corresponded with the image by *translation* only; this is a reasonable assumption for many imaging situations; errors in other parameters would only affect the registration stage of our system as described below.

Figure 1 shows a flowchart of our change detection approach. We assume that a site model of the scene is available and includes cultural features of interest. The site model and other previous information are said to be stored in a *site folder* with which new image (or images) are to be compared with. The change detection process consists of the following major steps:

- *Site Model to Image Registration*: The first step is to register the new image(s) to the model(s) contained in the site folder. Our system uses line feature matching globally to compensate for translational errors.
- *Site Model Validation*: This step verifies whether the objects in the site model are present in the new image by comparing predicted features with observed features. A confidence value is computed for each object in the model; low confidence values are likely to represent possible changes to the objects.
- *Structural Change Detection*: In this step, we analyze in more detail possible change in the site indicated in the previous step, and determine if the missing correspondences can be explained by the imaging and viewing conditions or whether evidence exists for actual changes. Our system is able to detect missing (or mis-placed) buildings, buildings with dimensional changes, and new buildings under certain conditions.
- *Site Model Updating*: In this step, 3-D models of changes are constructed where possible. These can be reported to a human analyst and reflected in an updated site model (which can then be used to process new images at the next cycle).

The following sections describe the processing at each step and illustrate with an example. More results and evaluations are provided in section 6. Our system has been tested primarily on data available for the Ft. Hood, Texas, site. The site models for our tests were constructed by using tools provided in the Radius Common Development Environment (RCDE) [6, 7]. The kinds of changes we are looking for occur over relatively long periods of time; unfortunately, we were not able to acquire data reflecting such changes for this site. Instead, we have modified the site models which should have the same effect as our system only compares images with site models rather than previous images. This method also provides a check on the accuracy and validity of previous site models. An earlier, shorter, conference version of this paper may be found in [8].

2 Site Model to Image Registration

The first step is to register a site model to an image. In our task, it is reasonable to assume that the imaging parameters are known to some accuracy and that the errors are such that if a site model is projected by the known parameters, its features will correspond with those from the image except for translational errors (which may be quite large, in the order of tens of pixels). The registration problem is then that of determining this translation, which we model as being uniform across the image.

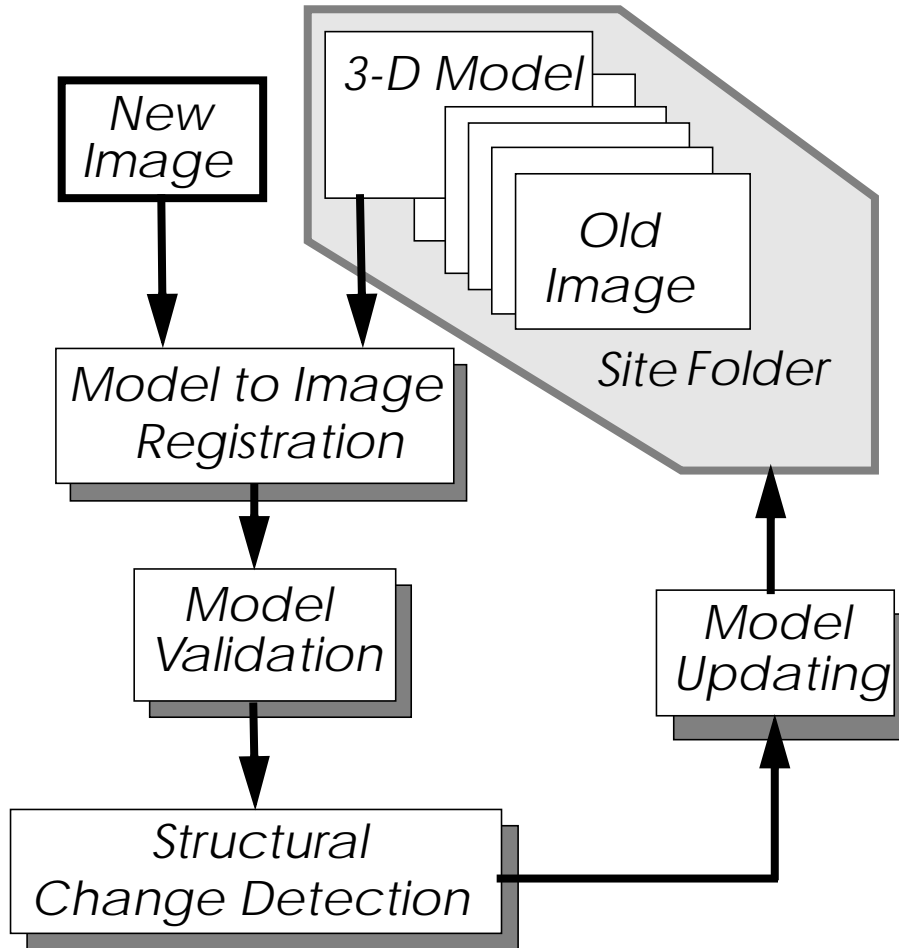


Figure 1. Flowchart of the change detection system

We need to decide what features of the models and image should be matched to determine the translation. The models are abstract, 3-D wire frame structures, the image is a 2-D array of intensity values. As the two levels of representation are not the same, we either need to project the models to a lower level or compute higher level representations from the image (or both). We do not think it is realistic to construct an image from the site models that can be put in a pixel-by-pixel correspondence with the real image (in addition to the difficulty of producing realistic intensity values, the site model is typically not complete). The task of constructing 3-D wire frame structures from the image is also difficult (though necessary for some of the steps discussed later). We have chosen to match lines extracted from the image with the lines projected from the site models by the known (approximately) camera geometry.

The line matching technique is adapted from an earlier method [9, 10]. It consists of computing the match quality for various possible translations in the expected range. The matches are computed for nearby segments and a score is computed. The score is a function of how close the matching segments are and how much they overlap.

We have a *candidate* pair of matching segments (one from the image set and one from the model set) when the two segments overlap, that is, a segment end points project inside the other segment (see figure 2). The segments also must lie within a certain “distance” of each other and the “angle” between the segments should be small. The calculation of this distance is as follows: If the two seg-

ments intersect, then the distance is zero. Otherwise the distance is the smallest projected distance of each segment end on the other segment, if it falls inside that segment.

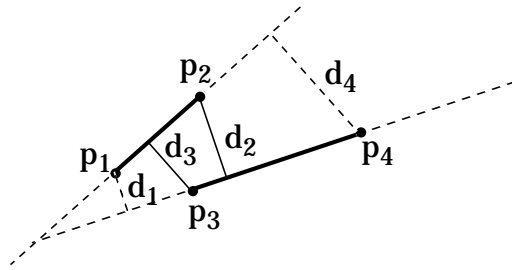


Figure 2. Distance between two segments p_1p_2 and p_3p_4 . Each segment end is projected onto the other segment. The distance is the minimum of the projection distances, if the projected point is inside the segment. In this case p_1 and p_4 project outside the other segment, therefore: $d(p_1p_2, p_3p_4) = \min(d_2, d_3)$.

Each pair of matching segments “votes” for a two-dimensional area in translation space (figure 3a). This accumulation mechanism in parameter space is a Hough-like transform [11] adapted to line segments. The two coordinates in the parameter space correspond to the x and y components of the translation. The amount of the contribution varies as a function of the distance between the two segments and the differences of segment orientation and length (figure 3b). We compute votes for matching pairs and add their contribution (figure 3c) into an accumulator array (parameter space). This contribution is designed to minimize noise effects as the vote is cast not only at the point corresponding to the translation between the two segments (the translation between the two segment centers (x_c, y_c)), but in a rectangular region (footprint) around these points, as follows:

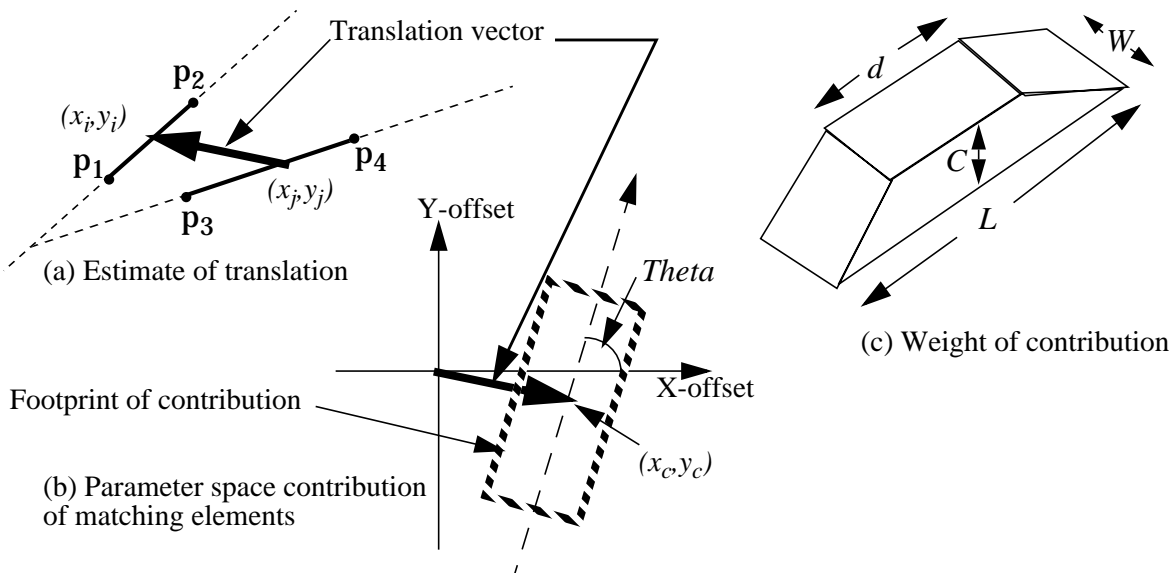


Figure 3. Computation of contribution from a pair of matching segments.

Let l_i^m and l_j^s represent the lengths of matching model segment i and scene segment j , and θ_i^m and θ_j^s their orientations, respectively.

- Center of footprint, $(x_c, y_c) = (x_i - x_j, y_i - y_j)$

- Orientation of footprint, $\theta = \frac{l_i^m \cdot \theta_i^m + l_j^s \cdot \theta_j^s}{l_i^m + l_j^s}$, where
- Length of base of contribution, $L = \max(4, l_i^m + l_j^s)$
- Length of top of contribution, $d = \min(l_i^m + l_j^s)$
- Width of contribution, $W = 3 \text{ pixels}$
- Height of contribution, $C = \left(1 + \frac{\min(l_i^m, l_j^s)}{\max(l_i^m, l_j^s)} \right)^2$

At the end of the voting process, a peak detected in the accumulator array gives the position of the best translation between the two sets of segments.

Figure 4 shows a portion of the site model including buildings in the Fort Hood Site. The site model is projected to according to the image viewpoint parameters and a set of line segments representing the model objects, visible in the image, is constructed. Figure 5 shows the line segments extracted from a portion of an image of the same site. Figure 6 shows the match values as a function of the displacement. The peak of this array gives the translation that best corresponds the model and image features; for the given example, it provides a good registration between the two. The model registered with the image is shown in Figure 7. We find this process to be quite robust, whether applied to small windows containing just a few buildings and to very large windows containing many tens of buildings (and other structures which may not be in the site model).

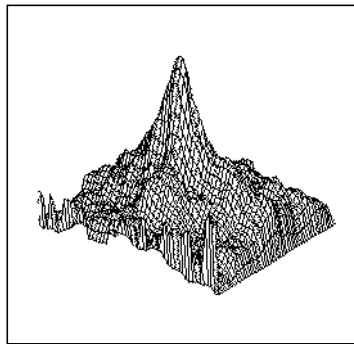


Figure 6. Accumulator array

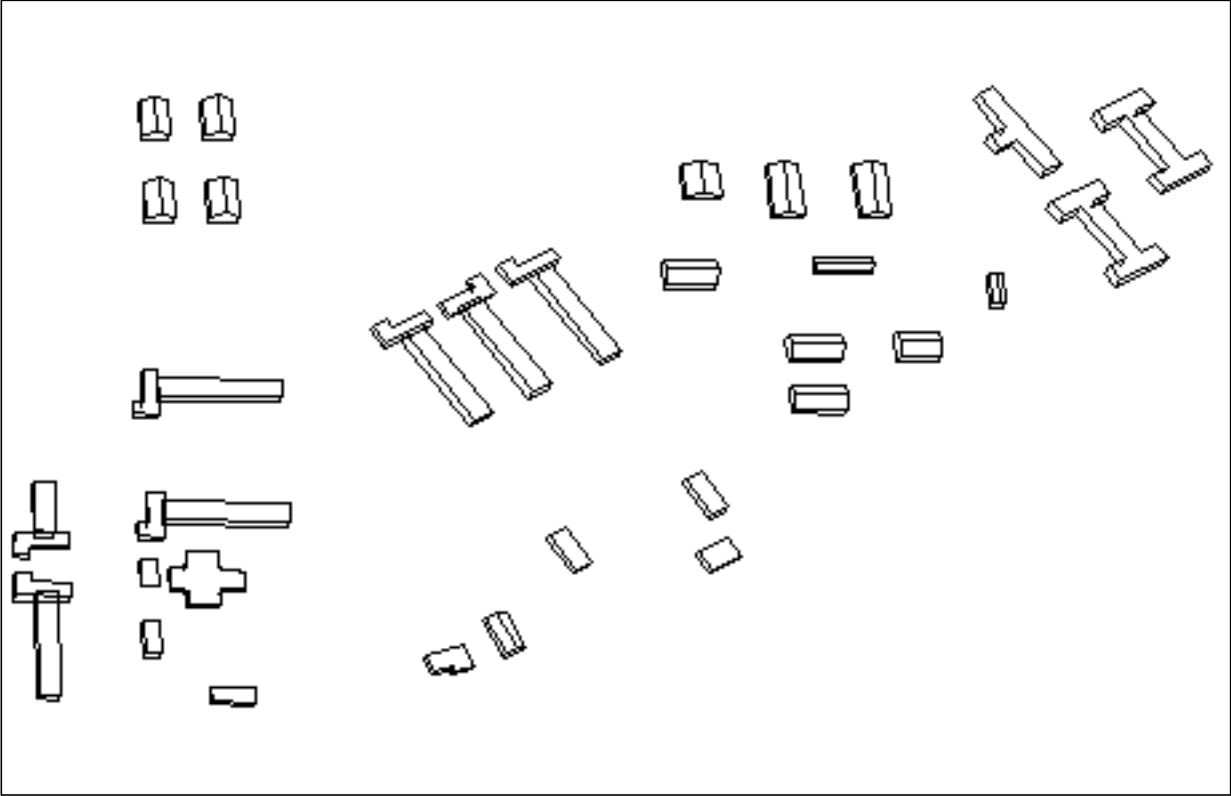


Figure 4. Portion of a Site Model

3 Site Model Validation

The purpose of model validation is to verify whether the objects in the site model are present in the new image in the same form or whether they should be examined in more detail for evidence of significant changes. The previous step of model and image registration helps provide a correspondence between the two at line segment level. For model validation, we combine evidence from a variety of object features such as lines and junctions. We also look for 3-D evidence; in a single image, this is done by evaluating expected shadows cast on the ground; in multiple images, this could come from feature correspondence information. Note that not all the features of the model may be visible in the image, some will be missing due to self and mutual occlusion. These occlusions can, however, be predicted from the viewing geometry and accounted for. There will also be missing evidence due to difficulties of feature extraction in images: low contrast edges may not be detected and line segments fragmented due to surface and ground texture.

We evaluate five kinds of evidence: edge visibility, edge presence, edge coverage, junction presence and shadow presence (these terms are explained in section 3.3). The confidence values derived take into account only visible elements from the particular viewpoint of the image after accounting for self and mutual occlusion. Before confidence values are calculated, the system deals

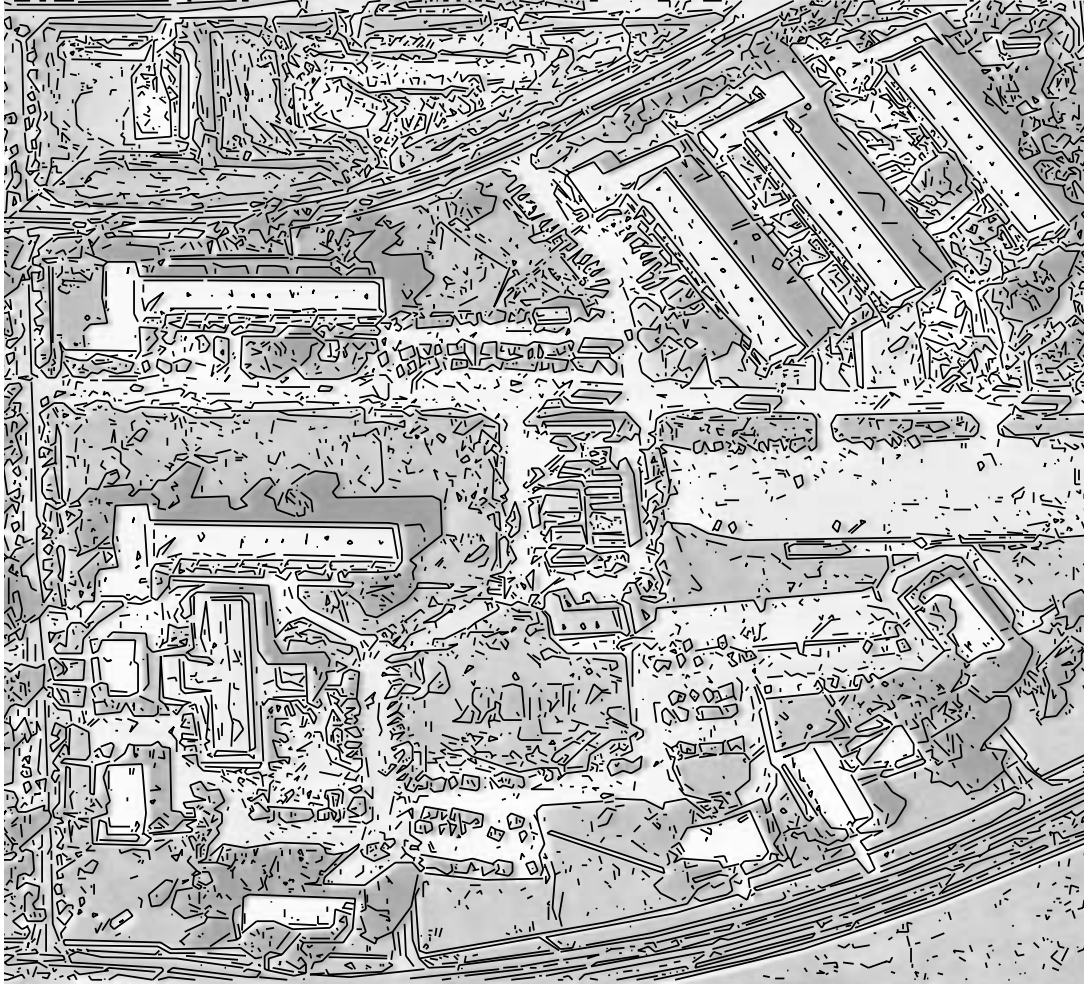


Figure 5. Line segments extracted from a portion of an image from Fort Hood, Texas

with a number of problems and ambiguities that need to be resolved, all inherent to any matching process, namely missing and multiple matches and coincidental alignments. We discuss these next.

3.1 Missing Features

To validate a model accurately we need to study the source of missing model-to-image correspondences. Some missing image features will be due to viewing conditions such as self-occlusion, occlusion by other objects, self shadows and shadows cast by nearby objects. These, however, can be predicted and explained from the site model itself. Missing correspondences may be due to over- or under-modeling of objects (Figure 8 and Figure 9) and are more difficult to predict from the model. The confidence associated with over- or undermodeled objects may thus be underestimated or difficult to calculate.

Over-modeling is due to the use of modeling primitives that introduce elements that do not correspond to actual physical elements or boundaries. Figure 8 shows a building that has been modeled by two rectangle parallelepipeds. The thick lines represent portions of the elements on the building model that do not correspond to physical boundaries. These can not be matched and the missing correspondences result in lower confidence.

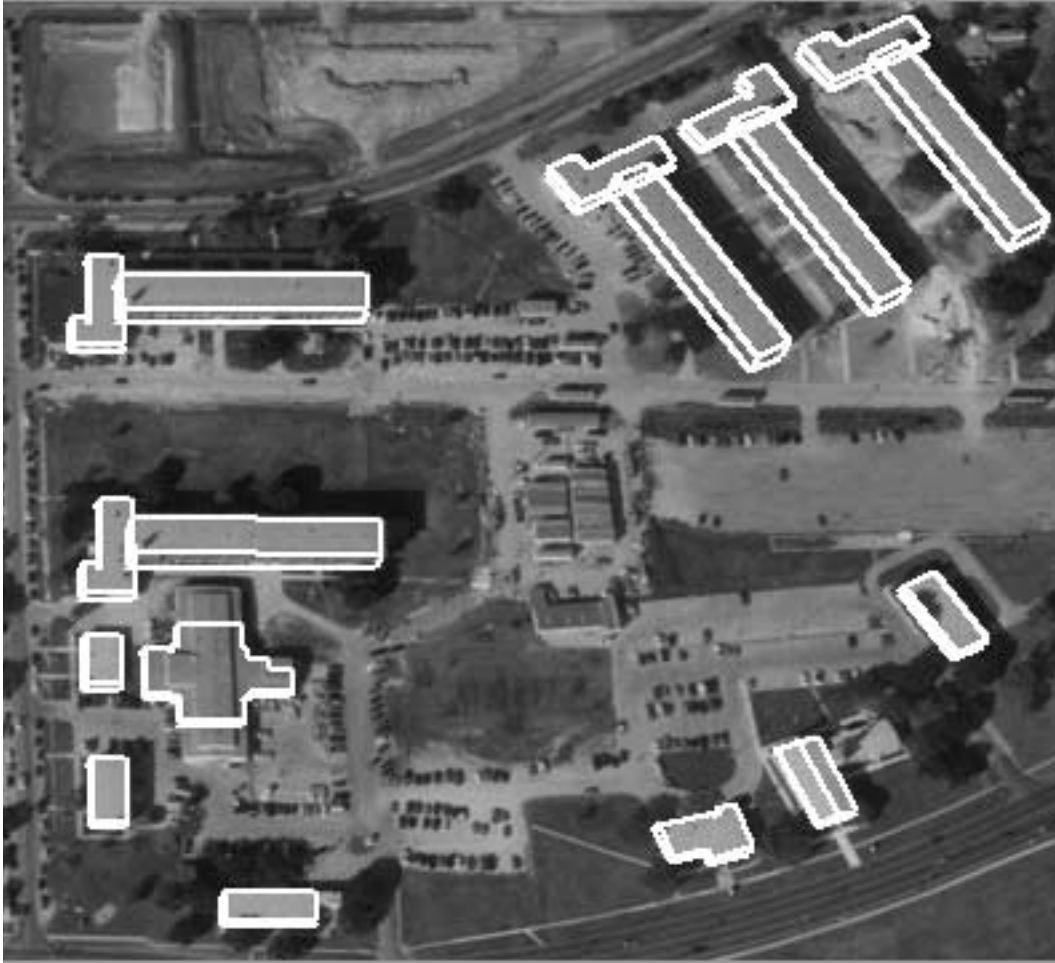


Figure 7. Site model registered with image

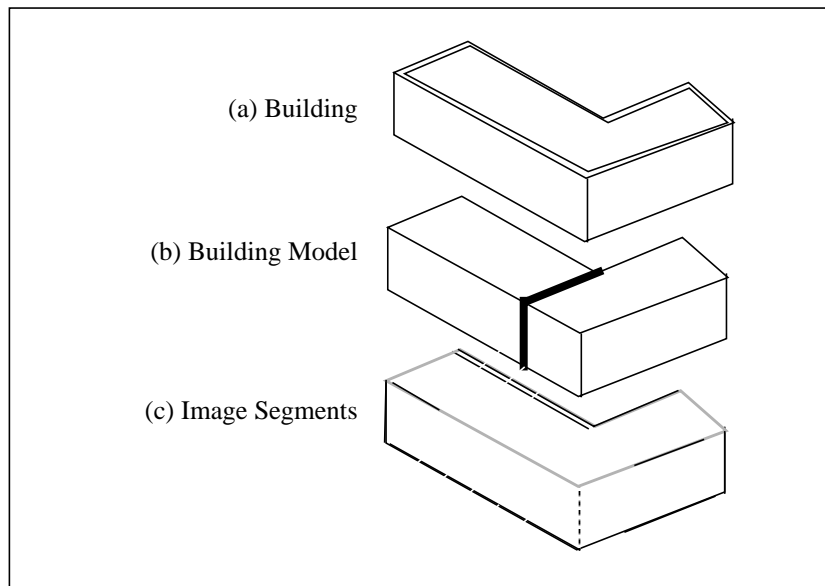


Figure 8. Missing match due to over-modeling.

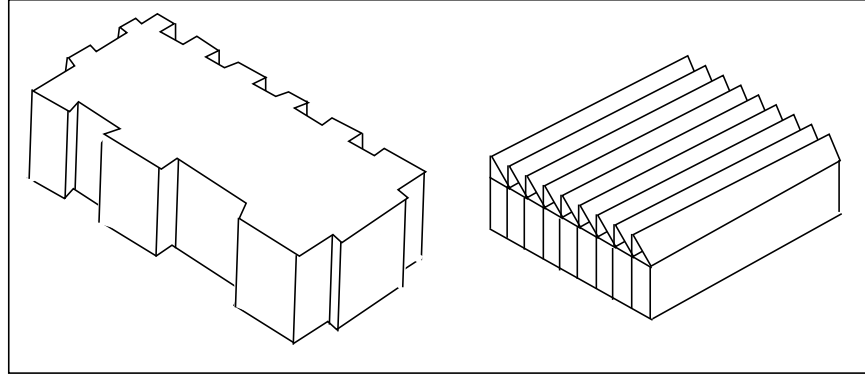


Figure 9. Some buildings may be under-modeled

Figure 9 shows two buildings that are likely to be under-modeled (i.e. modeled by simpler shapes) due to their complexity. These require additional search strategies designed to look for additional and possibly fragmented evidence, such as a large number of vertical or horizontal edge elements. Our system is currently not capable of determining these conditions, and thus the confidence values may be underestimated. It is assumed that some of these conditions may require annotations in the site model to help the system process these appropriately.

3.2 Ambiguities in Matching

There are several ambiguities inherent to the matching process that need to be resolved during validation. Our system currently deals with two of these: multiple matches, and coincidental alignments due to viewpoint, illumination direction, or due to adjacent structures.

3.2.1 Multiple Matches

The model-to-image matcher in the system corresponds each model element with one or more image elements. This is necessary to deal with expected fragmentation in the image elements. Fragmentation is due to inadequacies in the feature extraction process and due to actual image content, such as occluding trees, road boundaries and shadows. This may result in one-to-many correspondences (Figure 10) possibly involving more than one object. If a model segment matches multiple colinear image segments, all the image segments are considered to represent image support. If a model segments matches multiple parallel image segments, the overlap among these is considered to represent image support.

3.2.2 Coincidental Alignments.

Some multiple matches are due to coincidental alignments of buildings elements with elements of other adjacent or nearby structures (Figure 11). Some of these include roads, walkways, lawns, shadows, loading ramps, large vehicles, sidewalks, etc. Aligned elements may be thus detected in the image as longer line segments than expected, or predicted by the corresponding model elements. Since model segments matching longer image segments indicate a possibility of change (extension of dimension), we examine these cases to handle possible alignment in two cases: by examining nearby shadows with knowledge of the direction of illumination, and by examining adjacent structures. Coincidental alignments occur often in denser urban environments and are due to nearby and adjacent structures. We explicitly look for these to declare alignment or to declare change in dimensions. An example of some ambiguities and alignments is shown in Figure 12. If

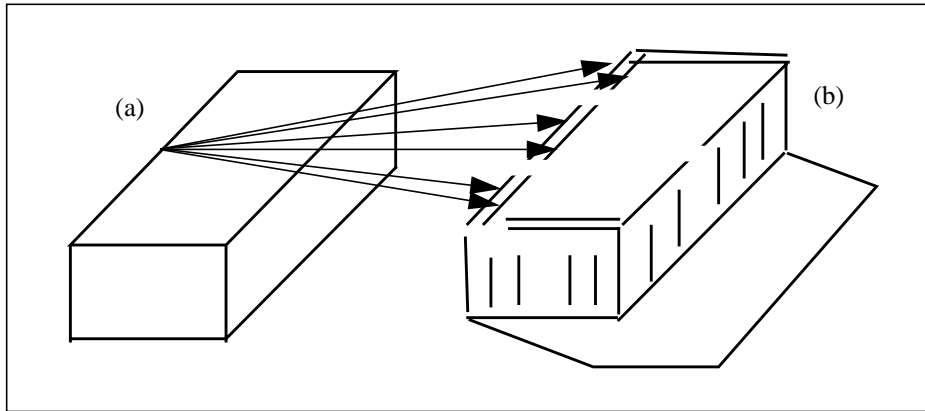


Figure 10. One-to-many correspondences.

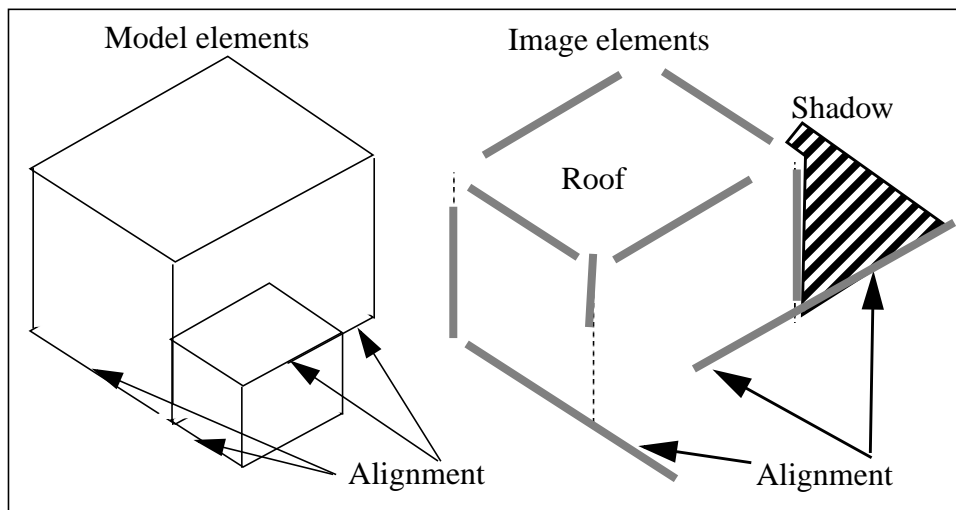


Figure 11. Coincidental alignments.

alignment is discovered, the extended portion, whether belonging to an adjacent structure or to a shadow boundary, is ignored.

3.3 Validation Confidence

We evaluate five kinds of evidence: edge visibility, edge presence, edge coverage, junction presence and shadow presence (these terms are explained below). Each evidence provides a score between 0 and 1 and a combined confidence score is computed by a linear weighted combination of them. We have chosen this method of combination for its simplicity. It works well in our tests but there may well be more optimal ways of combining such evidence. The confidence values derived take into account only visible elements from the particular viewpoint of the image after accounting for self and mutual occlusion. (Figure 13 and Figure 14):

Let x be a model object defined by a set of vertices and a set of edges. For each object, x , we calculate a confidence value $C(x)$ as a contribution of the following terms:

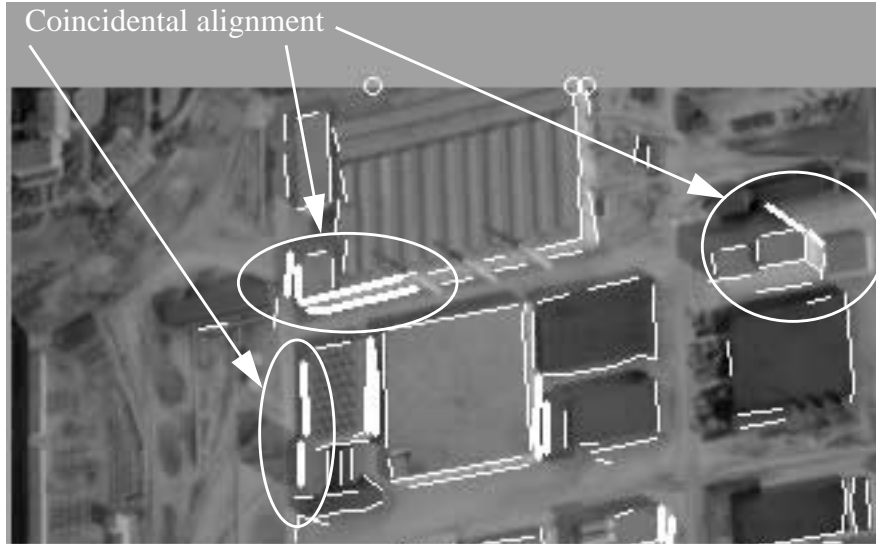


Figure 12. Example of ambiguities

Edge Visibility: $V(x)$ is given by the fraction of the model edges that are visible from the current viewpoint. Fewer visible edges result in lower confidence.

Edge Presence: $P(x)$ is defined as the fraction of the visible model edges that are matched to some image edges. In the schematic example shown in Figure 13 (a), all nine visible edges (dashed lines) have correspondences in the image (solid lines), giving a P value of 1.0. An object that is only 50% visible but that has the visible 50% corresponded to image edges has a P value of 1.0 also. P is calculated separately for **roof** elements, **vertical** wall elements and **base** wall elements which are given different weights (roof evidence is considered the most reliable, the wall base the least reliable).

Edge Coverage: $E(x)$ is defined as fraction of the lengths of the visible model edges that is actually covered by some image segments. Figure 13 (a) shows an object where all the model edges (dashed) have some, but small coverage; this object has good presence but poor coverage. Figure 13 (b) shows the opposite; a few model edges have good image edge support so the coverage is good but presence is not. $E(x)$ is also calculated separately for roof and wall elements. $E(x)$ is penalized by fragmented support. Fragmentation, $F(x)$, is defined as the ratio of the number of image segments to model to the number of model segments.

Junction Presence: $J(x)$ is defined as the ratio of the number of image L-junctions at locations predicted by the model (Figure 14) to the number of visible model vertices. Image junctions are extracted from the image from the line segments used for matching.

Shadow Presence: $S(x)$, is defined as the ratio of the number of potential shadow boundaries and junctions extracted from the image over number of visible shadow elements (boundaries and junctions) derived from the model (Figure 14). The image segments are labelled as potential shadow segments by noting the consistency of the “dark” side of the segment with respect to the direction of illumination. Segments oriented parallel to the projection of the direction of illumination correspond to possible shadow lines cast by vertical edges. The L-junctions formed (allowing for gaps) by potential shadow lines are labeled potential shadow junctions. Details on shadow evidence extraction may be found in [12].

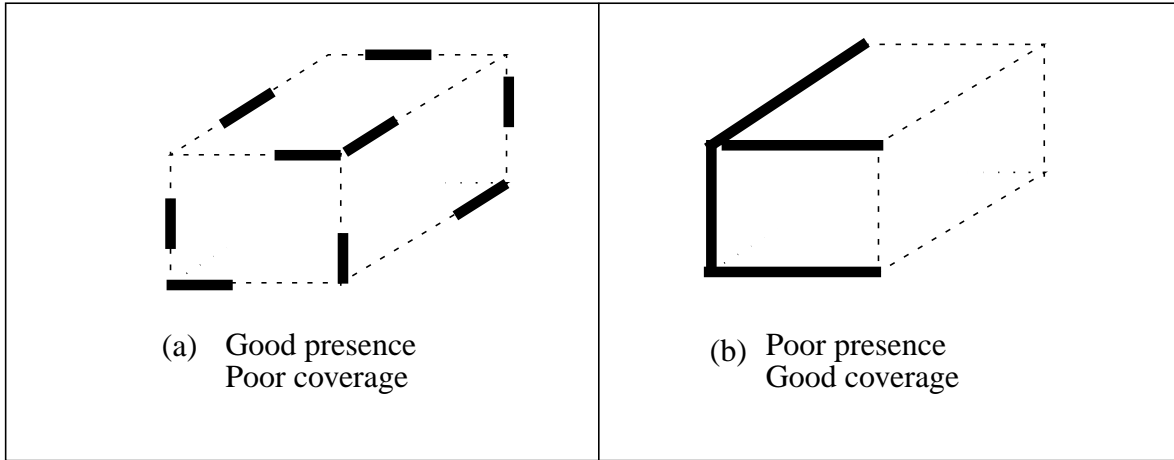


Figure 13. Presence and Coverage.

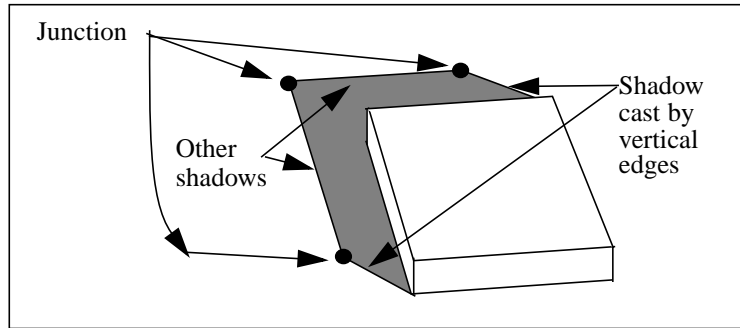


Figure 14. Shadows cast by a rectangular

A combined confidence value $C(x)$ is calculated by a linear weighted sum of the above described pieces of evidence, with $w_p = 7$ (roof), 5 (wall), and 3 (base), $w_v = 1$, $w_s = 3$, $w_j = 3$ and $w_m = w_p$:

$$C(x) = \frac{1}{\sum_i w_i} \cdot [w_p \cdot P(x) + w_v \cdot V(x) + w_s \cdot S(x) + w_j \cdot J(x) + w_m \cdot \frac{E(x)}{F(x)}]$$

High confidence values indicate good image support while low values denote low image support. Low values may signify change as lack of image support may be due to missing buildings, or buildings that have undergone significant change with respect to their current model. However, model buildings that have strong image support, may have changed also as additions to structures, such as a new wings, may not affect significantly the appearance of the previously modeled portions.

Figure 15 shows the results of the validation step applied to the image shown in Figure 7 earlier. The labels indicate the validation confidence level as a function of C values: high (**H**) for $C \geq 0.5$; medium (**M**) for $0.4 \leq C < 0.5$ and low (**L**) for $C < 0.4$. Note that a building indicated as having low (**L**) confidence has actually drastically changed (or was grossly mis-modeled) whereas the ones marked high (**H**) are in fact, unchanged. Medium level (**M**) typically denote buildings with moderate or “acceptable” image support. These assignments are arbitrary however, and would have to be set as a function of the task at hand. Some applications may require detailed explanations of possible change that require higher discrimination. Here we show only three for simplicity.

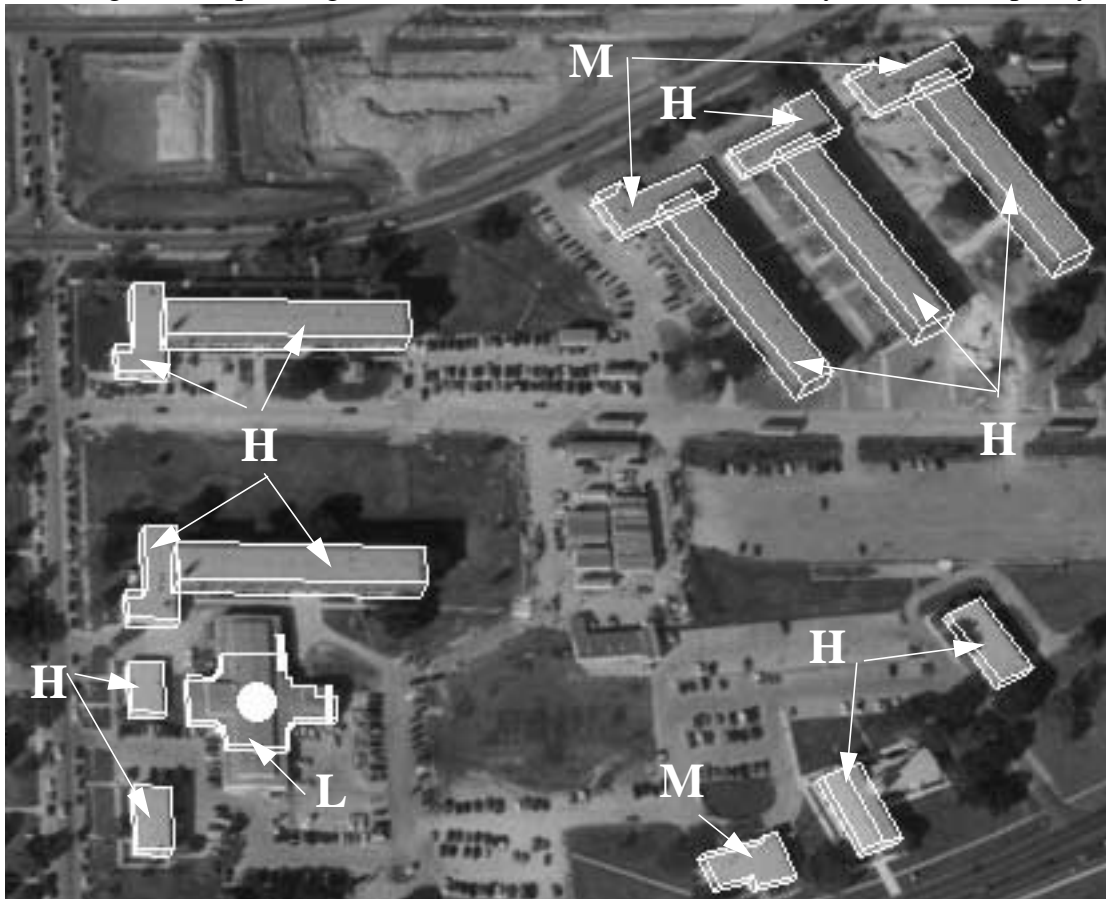


Figure 15. Validation result and labeled confidence levels.

4 Structural Change Detection

The validation step (validation) makes available information that is used to start analyses to determine structural changes. Two cues are used to investigate structural changes:

Validation Confidence values: These values reflect image support for a model object. Although low support may be due to poor image quality, lack of contrast, occlusion, or viewpoint, they can signify missing structures, substantially altered structures or incorrect modeling. Medium level support denote “acceptable” indication of presence with reduced support due to poor image quality, lack of contrast and other image dependent characteristics. High values clearly denote strong presence and image support, at least for the modeled portions.

Extra Image Elements: Model elements that correspond to image elements having greater extent (size) than that of the model elements provide preliminary indication of possible changes in the dimensions of the object.

The above cues are used to further analyze whether one of the following three classes of structural change has occurred: missing (or mis-placed) buildings, dimensional changes, and new buildings. The methods to infer these changes are described next.

4.1 Missing Buildings

Model buildings having very low confidence values denote poor image support. The possible causes for this condition are that either the model is incorrect, the structure is heavily occluded or that the building has been removed or destroyed (assuming that images are of sufficient quality), or that its position is grossly incorrect. A low confidence is sufficient to report a missing building, if additional images were available, they could be examined for confirmation. Figure 16 shows the result in a small window containing two modeled buildings. The model building on the right was added to the model by hand to test this condition. It is reported as having low confidence correctly as evidence for its presence can not be found in the image. Other examples of this condition are shown later in section 6.

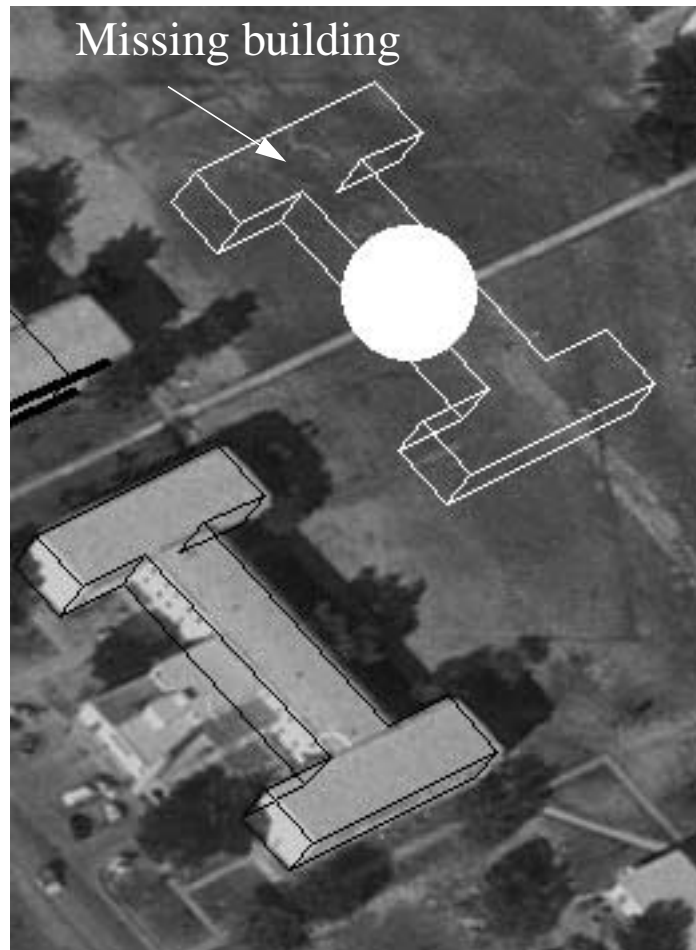


Figure 16. Missing building (white outlines)

4.2 Validated Buildings

Model buildings having moderate (**M**) to high (**H**) levels of support are considered validated. That is, their presence in the image is verified. The buildings labeled (**M**) typically require verification in another image to increase, if possible, their confidence level. An example is shown in Figure 17. The L-shaped building in the small window corresponds to the L-shapes building, labeled **M** on the top middle of the image shown in Figure 15. From this viewpoint the confidence value of this building increased to 0.5661, thus becoming validated with high confidence.

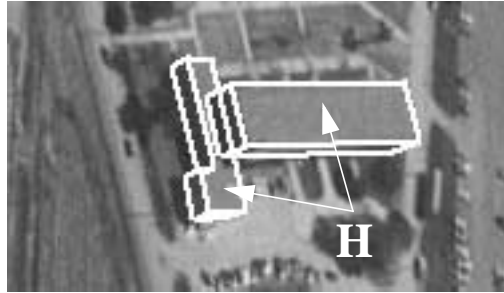


Figure 17. Validation of L-shaped building from another viewpoint.

4.3 Dimensional Changes to Modeled Structures

In some cases, regardless of confidence, the system is able to cue possible structural changes based on the fact that model edges are matched to **longer** image edges. Those cases where these conditions arise due to accidental alignments have been handled earlier, as explained in section 3.2. The remaining cases therefore represent cues for possible change. We can apply a building finding tool at this stage to confirm a change (and to derive a description of the change) as shown later in section 5, however, this relies on the building finder's ability to find new buildings. A less stringent criterion may be to only search for some additional evidence such as the extra edges casting a shadow and/or having other evidence of being above ground (perhaps by using multiple images). We have not implemented such partial analysis though components of it are available in the building finding tool.

Figure 18 (a) and (b) show two examples of evidence that support cueing dimensional changes. The matching and fine registration step correctly registers the modified models to the structures in the image. The thick white gray lines are the extended image segments that matched the corresponding model edges thus triggering the cue.

4.4 New Buildings

One important type of site change is the introduction of new structures. For such changes, the previous model is less useful but can still be relied on to provide some context. Such context can consist of areas of interest and characteristics of existing buildings (to check if buildings similar to existing ones have been constructed). Our system only uses the site model to mask out the modeled areas and a building finding tool is applied to all other areas, or all other areas containing large number of unmatched image features such as corners. The camera models and terrain models associated with the site are used to derive viewpoint and illumination parameters automatically. An

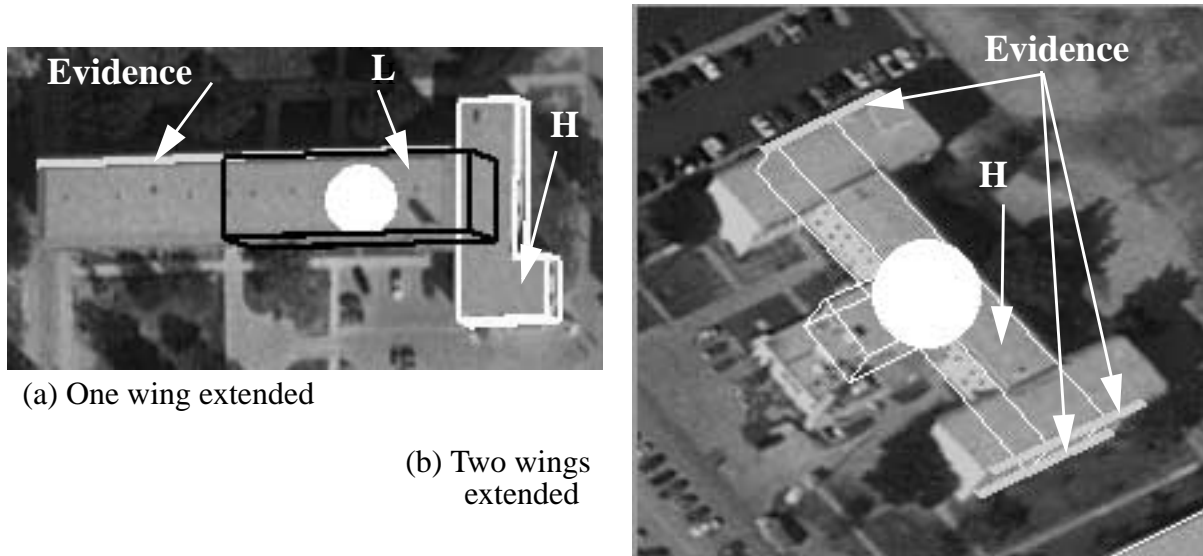


Figure 18. Actual change in dimensions.

example of this task is the next section (Figure 25). We have experimented with focus of attention mechanisms to select the areas where automated detection should be applied.

5 Model Updating

The next task is to make a model for the detected structural changes and to incorporate it in the site model. We describe two situations: where changes are made to existing structures and where new structures are detected.

5.1 Modeled Buildings

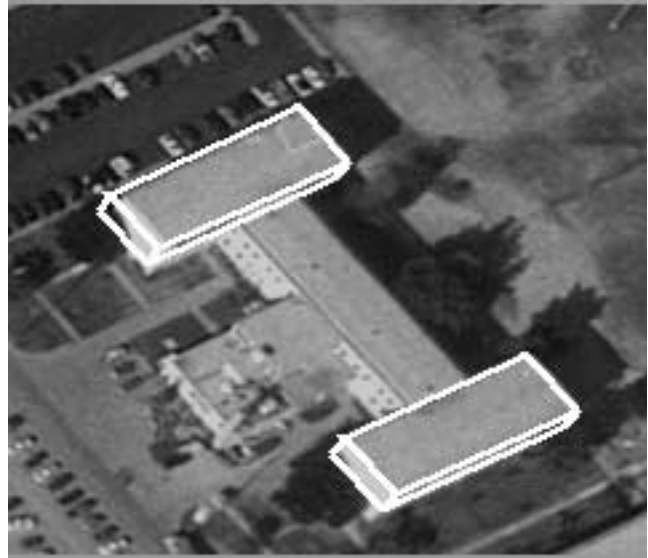
Changes in the dimensions of modeled structures that have been cued by the previous step need to be analyzed further, possibly using more than one view of the scene, if available. We use a monocular building detection system [12] to return the highest rated building hypothesis that can be formed in the location of the cued change. These are shown in Figure 19 for the two examples shown earlier in Figure 18. In this examples the image locations and sizes were selected manually before application of the automatic detection system. The 3-D models are however derived automatically (using shadow and wall evidence for 3-D inference).

5.2 New Buildings

Detection of new buildings is a more difficult task as the site model is less helpful. Our method consists of simply applying a building finding tool to areas of interest and reporting the results that are of sufficiently high confidence. Figure 20 shows the result of applying a monocular building detection system [12] to look for change in the form of new buildings (shown with white outlines). The areas modeled are ignored by the system. Typically the system would be instructed to locate new buildings in designated areas that are of interest, such as functional areas. The three buildings shown in white outlines are detected automatically and become suggested candidates to be added to the site model.



(a) 3-D model generated by building detector for the extended wing.



(b) 3-D models suggested for changed wings.

Figure 19. Suggested updating of cued changes

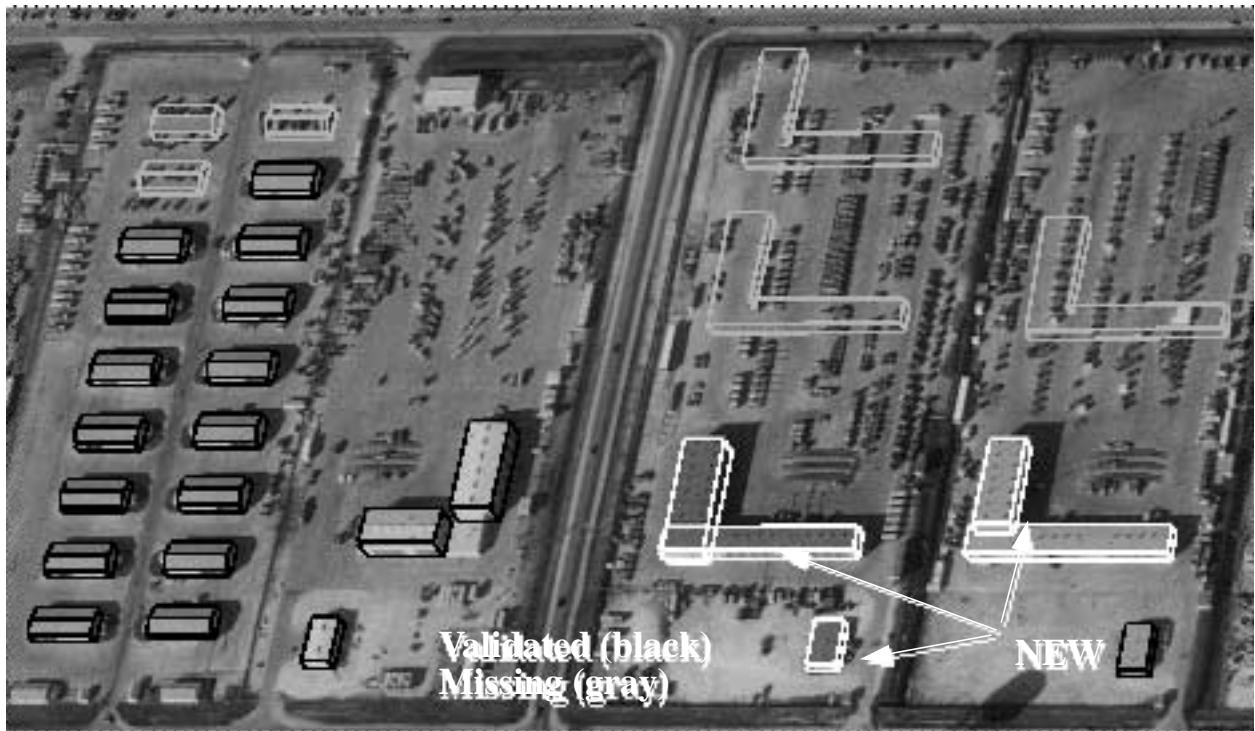


Figure 20. Model updating. New buildings are detected automatically

6 More Validation and Change Cueing Results

We show some additional examples and results in this section. Figure 21 shows an image of the Fort Hood site with the site model overlaid on it. The image size is 7775x7720 pixels, and the 3-D site model contains 79 objects representing building structures. In this example, the image and the

model are registered within a small translation. The system therefore is applied to each model object separately using small image windows. Processing time is about 15 seconds per structure on a Sun sparc-10 workstation, running under the RCDE (Radius Common Development Environment.) The results are summarized in table 1. It shows the number of building objects visible in the image and the distribution of validation confidence values (the label codes in the table are in reference to the graphical results shown below in Figure 22 through Figure 25. **H**, **M** and **L** denote high, medium and low validation confidence respectively). The confidence values are dependent on the image content and may not necessarily reflect structural changes but in this example at least, there is a high correlation between confidence level and the number of buildings changed, not changed or missing in the rest of the table. All matching ambiguities, with one exception, shown in Figure 22, are correctly handled. This case involves an alignment with a ground feature not present in the model, a situation, not currently handled by the system. Of the 54 non-changed buildings, one is cued incorrectly to have changed. This building is shown on the left of Figure 22, and is the same one for which the unresolved coincidental alignment appears to indicate that the building has been extended towards the left. This situation is however likely to be corrected by confirmation of the change using another view of the building.

Fourteen buildings that are actually present in the image had changes. Thirteen of these are found to be changed correctly: Three in Figure 22, five in Figure 23, three in Figure 24 and two in Figure 25. These are shown in the figures with a circle on top and thick white lines cueing evidence of dimensional changes. The remaining changed building is the L-shaped building on the right of figure23. The building has two wings, both of which have changed. Only the left wing is detected to have changed. This situation is likely to be corrected by reconciling the output from more than one view.

Buildings that change considerably or are missing have poor image support, resulting in low validation confidence (the buildings labeled **L** in Figure 22 through Figure 25). There are 12 of these, 11 of which were added by hand to test the “missing building” detection capability. The remaining one, represents a significantly changed building (The cross-shaped building in Figure 24). All these are labeled correctly as changed or missing.

Table 1: Summary of Results

Image (fhov927)	Visible Buildings	Validation Confidence			Non-changed Buildings			Changed Buildings			Missing Buildings		
		High (H)	Medium (M)	Low (L)	Number of buildings	Reported non-changed	Reported changed	Number of buildings	Reported changed	Reported non-changed	Number of buildings	Reported missing	Validated
No.	79	54	13	12	54	53	1	14	13	1	11	11	0
%	100	68.3	16.4	15.1	100	98.1	1.9	100	92.8	7.3	100	100	0.0

7 Choice of Parameters

Our system uses several parameters in its decision making processes at various levels. Choice of these parameters, of course, determines the quality of the results that are obtained even though we have attempted to make the system not very sensitive to them. Ideally, the parameter values should be based on a mathematical analysis of the algorithms and be a function of the parameters of the input images and of any known parameters of the site. Unfortunately, such an analysis is difficult due to the complexities of the algorithms and estimating appropriate image parameters. In our system, we make decision parameters a function of the image parameters that are supplied with the images, such as the resolution. Additional refinements may be possible by using context of the site. We have set the parameters by an informal analysis of the process and testing with available sets of data. All of our examples use the same parameter values.

8 Conclusion

We have shown some results and capabilities of our system for detecting and describing structural changes. It has been tested on real images (though with simulated changes to the model) and seems to be quite effective at finding significant changes in rather complex images. It is able to find missing (or misplaced) buildings, buildings with changed (or incorrectly modeled) dimensions and new buildings (or previously unmodeled buildings). This system relies on the use of a single image to find changes. We anticipate that its performance would be significantly enhanced by use of multiple images as they would provide independent evidence of changes and also allow the reasoning to proceed more directly in 3-D space. Several multiple image building finders are becoming available [13, 14, 15] and could be easily incorporated in our method. Our system has been ported to an industrial laboratory for possible use in current applications.



Figure 21. Portion of an image from Fort Hood, Texas.

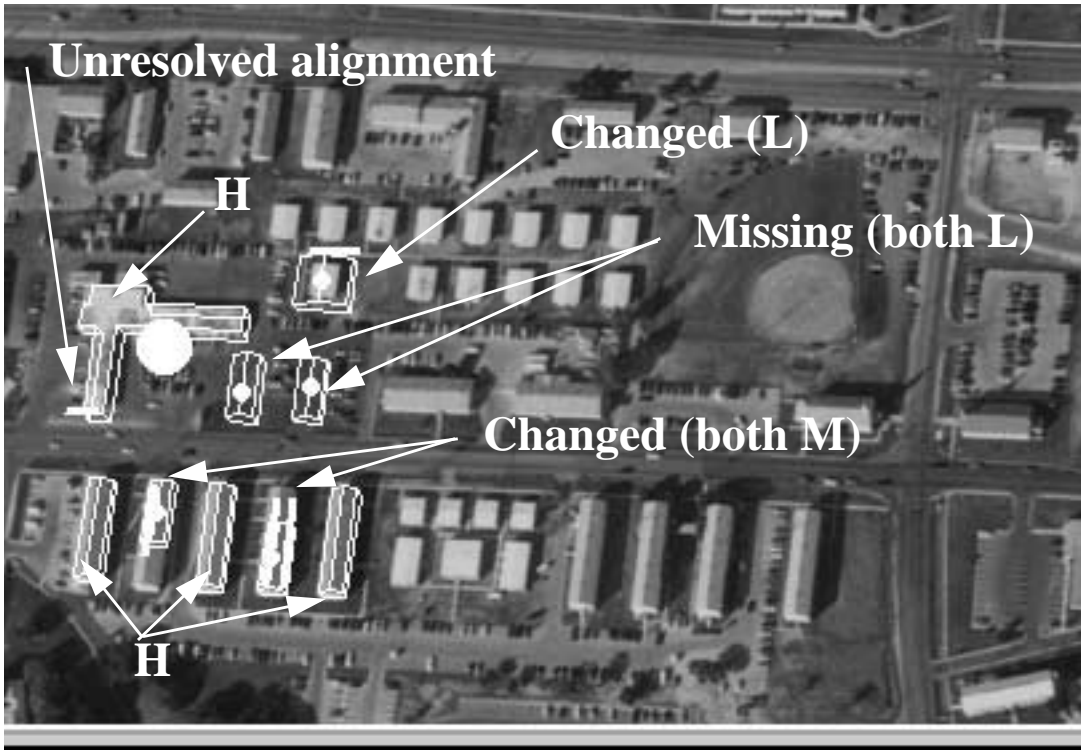


Figure 22. Validation and change detection result for area A of Fort Hood



Figure 23. Validation and change detection results for AREA B of Fort Hood example



Figure 24. Validation and change detection result for AREA (C of Fort Hood, Texas

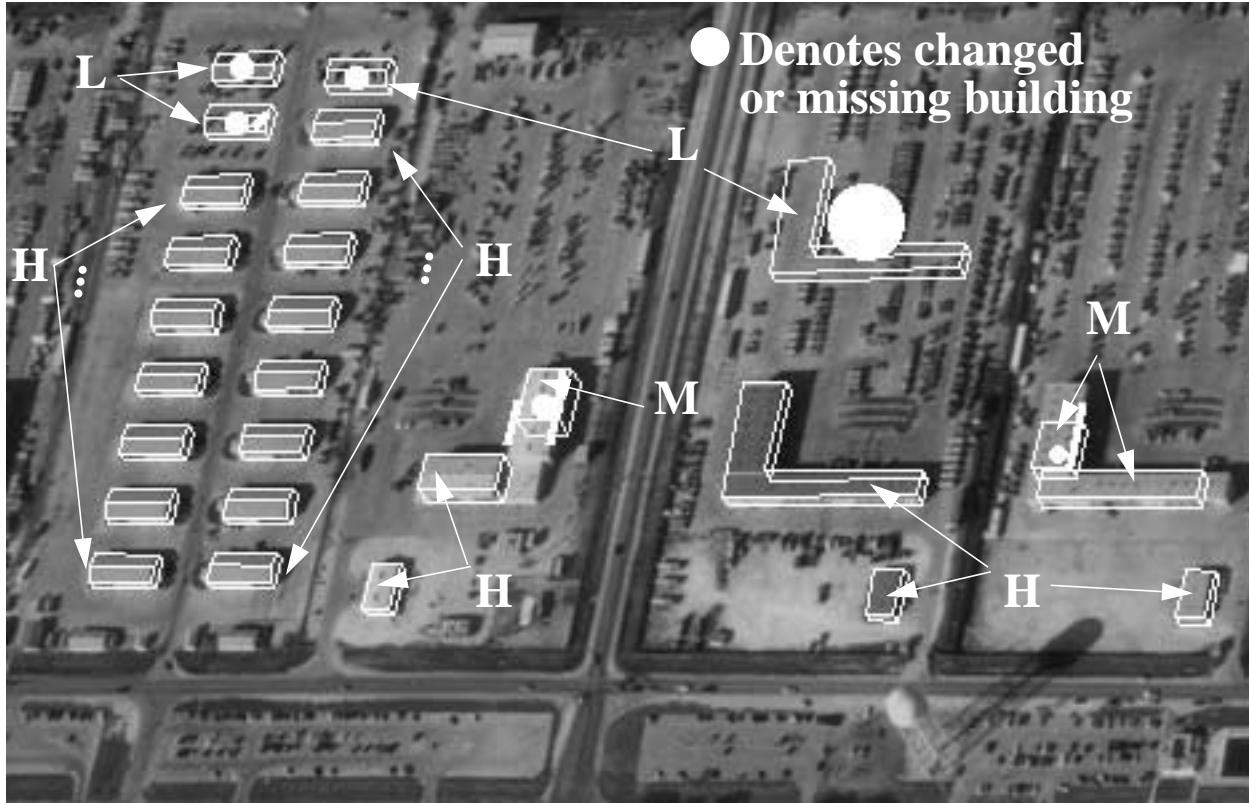


Figure 25. Validation and change detection result for AREC D of Fort Hood.

References

- [1] Lillestrand, R. L., "Techniques for Change Detection," *IEEE Transactions on Computers*, 7 (1972), pp. 654-659.
- [2] Khalaj, B., Aghajan H. and Kailath T., "Automated Direct Patterned Wafer Inspection," *Proceedings of the IEEE Workshop on Applications of Computer Vision*, Palm Springs, California (1992) , pp. 266-273.
- [3] Gerson, G. and Wood S., "The RADIUS Testbed System," in: Firschein, O and Strat, T, ed, RADIUS: Image Understanding for Imagery Intelligence, (Morgan Kaufmann Publishers, San Francisco, 1997), pp. 5-12.
- [4] Huttenlocher, D. and Ullmann S., "Recognizing Solid Objects by Alignment with an Image," *International Journal of Computer Vision*, Vol. 5, No.2., November, 1990, pp. 195-212.
- [5] Chellappa, R., Zhang X., Burlina P., Lin, C., Zheng Q., Davis L. and Rosenfeld A., "An Integrated System for Site Model Supported Monitoring of Transportation Activities in Aerial Images," *Proceedings of the DARPA Image Understanding Workshop*, Palm Springs, California, (Morgan Kaufman Publisher, San Francisco 1996), pp. 275-304.
- [6] Strat, T., Quam L., Mundy J., Welty, R., Bremner, W., Horwedel, M., Hackett, D. and Hoogs, A, "The RADIUS Common Development Environment," *Proceedings of the DARPA Image Understanding Workshop*, San Diego, California, (Morgan Kaufman Publisher, San Francisco, 1992), pp. 215-226.
- [7] Fua, P., "Cartographic Applications of Model-Based Optimization," *Proceedings of the Image Understanding Workshop*, Palm Springs, California, (Morgan Kaufman Publisher, San Francisco, 1996), pp. 409-419.
- [8] Huertas, A. and Nevatia R., "Detecting Changes in Aerial Views of Man-Made Objects," *Proceedings, IEEE Sixth International Conference on Computer Vision*, Bombay, India, (January, 1998), pp. 73-80.
- [9] Médioni, G., Huertas, A., and Wilson, M., "Automatic Registration of Color Separation Films," *Machine Vision and Applications*, 4 (Springer-Verlag, New York, 1991), pp. 33-51.
- [10] Huertas, A., Bejanin, M. and Nevatia, R., "Model Registration and Validation," in: Gruen, A., Kuebler, O. and Agouris. P., ed, Automatic Extraction of Man-Made Objects from Aerial and Space Images, (Birkhauser Verlag, Basel 1995), pp. 33-42.
- [11] Hough, P., "Methods and Means for Recognizing Complex Pattern,s" U.S. Patent 3069654, December 18, (1962).
- [12] Lin, C., Huertas, A. and Nevatia, R., "Detecting Buildings from Monocular Images," in: Gruen, A., Kuebler, O. and Agouris, P., ed, Automatic Extraction of Man-Made Objects from Aerial and Space Images, (Birkhauser Verlag, Basel 1995), pp. 125-134.
- [13] Noronha, S. and Nevatia, R., "Detection and Description of Buildings from Multiple Aerial Images," *Proceedings of the DARPA Image Understanding Workshop*, Palm Springs, California, (Morgan Kaufman Publisher, San Francisco, 1996), pp. 469-478
- [14] Jaynes, C., Stolle, F. and Collins, R., "Task Driven Perceptual Organization for Extraction of Rooftop Polygons," *Proceedings of the ARPA Image Understanding Workshop*, Monterey, California (Morgan Kaufmann Publishers, San Francisco, 1994), pp. 359-365.
- [15] Collins, R., Hanson, A, Riseman, E. and Schultz, H., "Automatic Extraction of Buildings from Aerial Images," in: Gruen, A., Kuebler, O. and Agouris, P., ed, Automatic Extraction of Man-Made Objects from Aerial and Space Images, (Birkhauser Verlag, Basel 1995), pp. 169-178.

Detecting Changes in Aerial Views of Man-Made Structures

Andres Huertas and Ramakant Nevatia*

Institute for Robotics and Intelligent Systems

University of Southern California

Los Angeles, California 90089-0273

{nevatia|huertas}@iris.usc.edu

www: iris.usc.edu

Abstract

An important application of machine vision is to provide a means to monitor a scene over a period of time and report significant changes. Comparing intensity values of successive images is not effective as such changes don't necessarily reflect actual changes at a site but might be caused by changes in the view point and illumination. We also want to ignore seasonal changes and focus on structural changes instead. We take the approach of comparing a 3-D model of the site, prepared from previous images, with new images to infer significant changes. This task is difficult as the images and the models have very different levels of abstract representations. Our approach consists of several steps: model to image registration aims to bring the site model and the images into close correspondence; model validation to confirm the presence of model objects in the image; structural change detection seeks to resolve matching problems and provide cues indicating possibly changed structures; model updating aims to suggest modified models for existing model structures and to incorporate new models. Our system is able to detect changes, such as missing (or mis-modeled) buildings, changes in model dimensions, and new buildings under some conditions. Several experimental results are presented.

Keywords: Structural change detection, image to model correspondence, model validation, building detection, aerial image analysis, vision application

* This research was supported by Contract No. DACA-76-93-C-0014 from the Defense Advanced Research Projects Agency (DARPA) and monitored by the Topographic Engineering Research Center of the U.S. Army.

1 Introduction

One of the key applications for aerial and space image analysis is in monitoring and keeping track of changes on the ground. This could be for various purposes such as urban planning, agricultural analysis, environmental monitoring and military intelligence. Currently, such analysis is conducted by human *image analysts*, however, this is a tedious and time consuming task for the humans and allows for only a small fraction of the available imagery to be examined. Many kinds of changes could be relevant such as in structures on the ground, in natural vegetation and in environmental conditions. We focus on changes in man-made structures such as construction or modifications of roads or buildings. Changes in the desired structures should be reflected in some changes in the image, not all changes in the images may be caused by 3-D structural changes. Image intensities can vary due to a number of factors such as changes in illumination, viewpoint, and atmospheric conditions. Further, seasonal variations may cause changes in vegetation and ground cover; while these represent “real” changes on ground, they may not correspond to structural changes.

There is little previous work in structural change detection. Early work in change detection was based on determining pixel intensity changes [1], which do not necessarily correspond to the desired changes as explained above. Structural change detection has similarities to the problems of industrial inspection where a new image is compared with a reference image (or a model) to detect significant variations (see for example, [2]); however, in industrial settings, viewpoint and illumination conditions are typically fixed and controlled. In outdoor scenes, we must deal with the environmental conditions as they are and may have only limited control over how the data gets collected. Thus, to find structural changes, we should not compare images taken at different times directly but rather compare new images (or descriptions derived from them) to an abstract model derived from previous observations; such models have come to be called *site models* [3]. This approach also allows an updating of the site models which may be one of the prime goals of the change detection analysis.

Detecting changes at a site based on abstract models from previous observations poses several significant challenges for computer vision techniques. A new image can not be directly corresponded to an abstract model. Instead, we must compute descriptions from the image that can be corresponded with the model or descriptions that can be derived from the model. This is a common problem in object detection in computer vision and various techniques such as *alignment* have been developed to solve it [4]. The change detection problem is simpler to the extent that some parameters of the *pose* of the objects may be known *a priori*. However, the objects themselves may have changed and not fit a prior model exactly. Also, aerial images typically contain a large number of man-made and natural objects, not all of which may have been modeled (for example, we do not assume models for trees in a scene). The objects of interests may be partially (or totally) occluded by other objects and shadows cast by them may cause confusion. The images also contain significant amount of texture, thus leading to a large number of features at the lower-levels (such as edges) that prohibit use of combinatorial techniques to search for desired objects. Finally, we need to verify that the suspected changes actually correspond to some 3-D structures and to derive a description of the changes where possible.

The system described in this paper is designed for detecting changes in 3-D building structures, though many of the techniques may also apply to other objects such as roads. We do not handle mobile objects, such changes are addressed, for example in [5]. We further assume that the build-

ings are rectilinear and that composite shapes (such as “L” or “T”) are described by their rectangular components, thus allowing us to have a uniform representation for the rectilinear shapes. Each rectangular part is represented by its 3-D wire-frame (consisting of 8 vertices and 12 edges). Our techniques should generalize to other shapes but some of the detailed procedures would have to be changed. We assume that the camera geometry and approximate viewpoint from which images are taken are known. Specifically, we assume that the errors are such that a projected model can be corresponded with the image by *translation* only; this is a reasonable assumption for many imaging situations; errors in other parameters would only affect the registration stage of our system as described below.

Figure 1 shows a flowchart of our change detection approach. We assume that a site model of the scene is available and includes cultural features of interest. The site model and other previous information are said to be stored in a *site folder* with which new image (or images) are to be compared with. The change detection process consists of the following major steps:

- *Site Model to Image Registration*: The first step is to register the new image(s) to the model(s) contained in the site folder. Our system uses line feature matching globally to compensate for translational errors.
- *Site Model Validation*: This step verifies whether the objects in the site model are present in the new image by comparing predicted features with observed features. A confidence value is computed for each object in the model; low confidence values are likely to represent possible changes to the objects.
- *Structural Change Detection*: In this step, we analyze in more detail possible change in the site indicated in the previous step, and determine if the missing correspondences can be explained by the imaging and viewing conditions or whether evidence exists for actual changes. Our system is able to detect missing (or mis-placed) buildings, buildings with dimensional changes, and new buildings under certain conditions.
- *Site Model Updating*: In this step, 3-D models of changes are constructed where possible. These can be reported to a human analyst and reflected in an updated site model (which can then be used to process new images at the next cycle).

The following sections describe the processing at each step and illustrate with an example. More results and evaluations are provided in section 6. Our system has been tested primarily on data available for the Ft. Hood, Texas, site. The site models for our tests were constructed by using tools provided in the Radius Common Development Environment (RCDE) [6, 7]. The kinds of changes we are looking for occur over relatively long periods of time; unfortunately, we were not able to acquire data reflecting such changes for this site. Instead, we have modified the site models which should have the same effect as our system only compares images with site models rather than previous images. This method also provides a check on the accuracy and validity of previous site models. An earlier, shorter, conference version of this paper may be found in [8].

2 Site Model to Image Registration

The first step is to register a site model to an image. In our task, it is reasonable to assume that the imaging parameters are known to some accuracy and that the errors are such that if a site model is projected by the known parameters, its features will correspond with those from the image except for translational errors (which may be quite large, in the order of tens of pixels). The registration problem is then that of determining this translation, which we model as being uniform across the image.

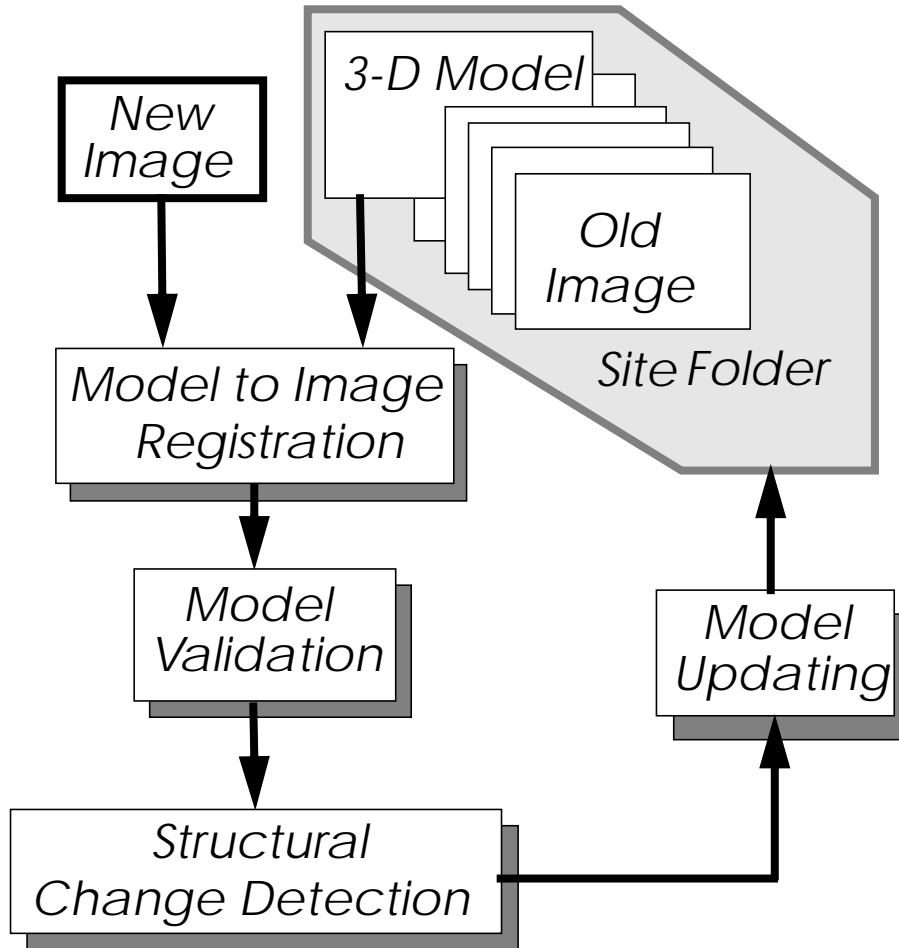


Figure 1. Flowchart of the change detection system

We need to decide what features of the models and image should be matched to determine the translation. The models are abstract, 3-D wire frame structures, the image is a 2-D array of intensity values. As the two levels of representation are not the same, we either need to project the models to a lower level or compute higher level representations from the image (or both). We do not think it is realistic to construct an image from the site models that can be put in a pixel-by-pixel correspondence with the real image (in addition to the difficulty of producing realistic intensity values, the site model is typically not complete). The task of constructing 3-D wire frame structures from the image is also difficult (though necessary for some of the steps discussed later). We have chosen to match lines extracted from the image with the lines projected from the site models by the known (approximately) camera geometry.

The line matching technique is adapted from an earlier method [9, 10]. It consists of computing the match quality for various possible translations in the expected range. The matches are computed for nearby segments and a score is computed. The score is a function of how close the matching segments are and how much they overlap.

We have a *candidate* pair of matching segments (one from the image set and one from the model set) when the two segments overlap, that is, a segment end points project inside the other segment (see figure 2). The segments also must lie within a certain “distance” of each other and the “angle” between the segments should be small. The calculation of this distance is as follows: If the two seg-

ments intersect, then the distance is zero. Otherwise the distance is the smallest projected distance of each segment end on the other segment, if it falls inside that segment.

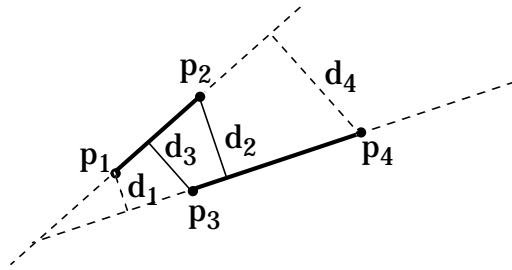


Figure 2. Distance between two segments p_1p_2 and p_3p_4 . Each segment end is projected onto the other segment. The distance is the minimum of the projection distances, if the projected point is inside the segment. In this case p_1 and p_4 project outside the other segment, therefore: $d(p_1p_2, p_3p_4) = \min(d_2, d_3)$.

Each pair of matching segments “votes” for a two-dimensional area in translation space (figure 3a). This accumulation mechanism in parameter space is a Hough-like transform [11] adapted to line segments. The two coordinates in the parameter space correspond to the x and y components of the translation. The amount of the contribution varies as a function of the distance between the two segments and the differences of segment orientation and length (figure 3b). We compute votes for matching pairs and add their contribution (figure 3c) into an accumulator array (parameter space). This contribution is designed to minimize noise effects as the vote is cast not only at the point corresponding to the translation between the two segments (the translation between the two segment centers (x_c, y_c)), but in a rectangular region (footprint) around these points, as follows:

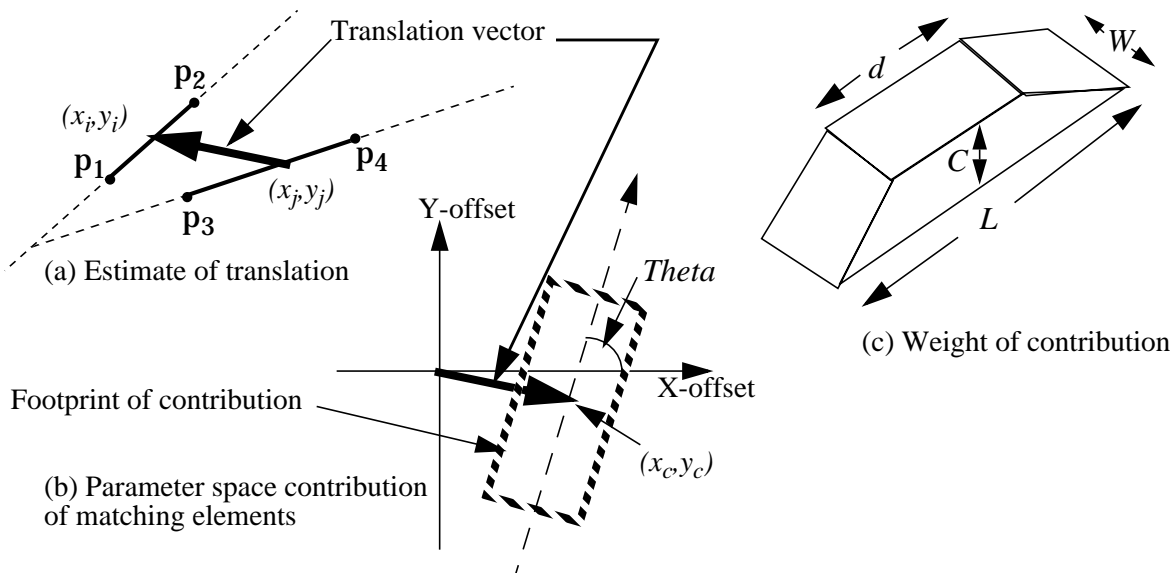


Figure 3. Computation of contribution from a pair of matching segments.

Let l_i^m and l_j^s represent the lengths of matching model segment i and scene segment j , and θ_i^m and θ_j^s their orientations, respectively.

- Center of footprint, $(x_c, y_c) = (x_i - x_j, y_i - y_j)$

- Orientation of footprint, $\theta = \frac{l_i^m \cdot \theta_i^m + l_j^s \cdot \theta_j^s}{l_i^m + l_j^s}$, where
- Length of base of contribution, $L = \max(4, l_i^m + l_j^s)$
- Length of top of contribution, $d = \min(l_i^m, l_j^s)$
- Width of contribution, $W = 3 \text{ pixels}$
- Height of contribution, $C = \left(1 + \frac{\min(l_i^m, l_j^s)}{\max(l_i^m, l_j^s)} \right)^2$

At the end of the voting process, a peak detected in the accumulator array gives the position of the best translation between the two sets of segments.

Figure 4 shows a portion of the site model including buildings in the Fort Hood Site. The site model is projected to according to the image viewpoint parameters and a set of line segments representing the model objects, visible in the image, is constructed. Figure 5 shows the line segments extracted from a portion of an image of the same site. Figure 6 shows the match values as a function of the displacement. The peak of this array gives the translation that best corresponds the model and image features; for the given example, it provides a good registration between the two. The model registered with the image is shown in Figure 7. We find this process to be quite robust, whether applied to small windows containing just a few buildings and to very large windows containing many tens of buildings (and other structures which may not be in the site model).

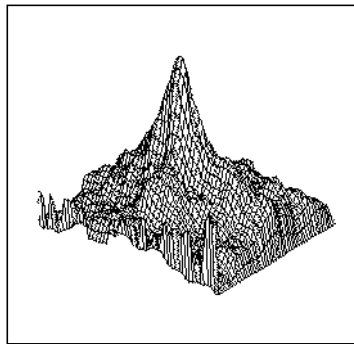


Figure 6. Accumulator array

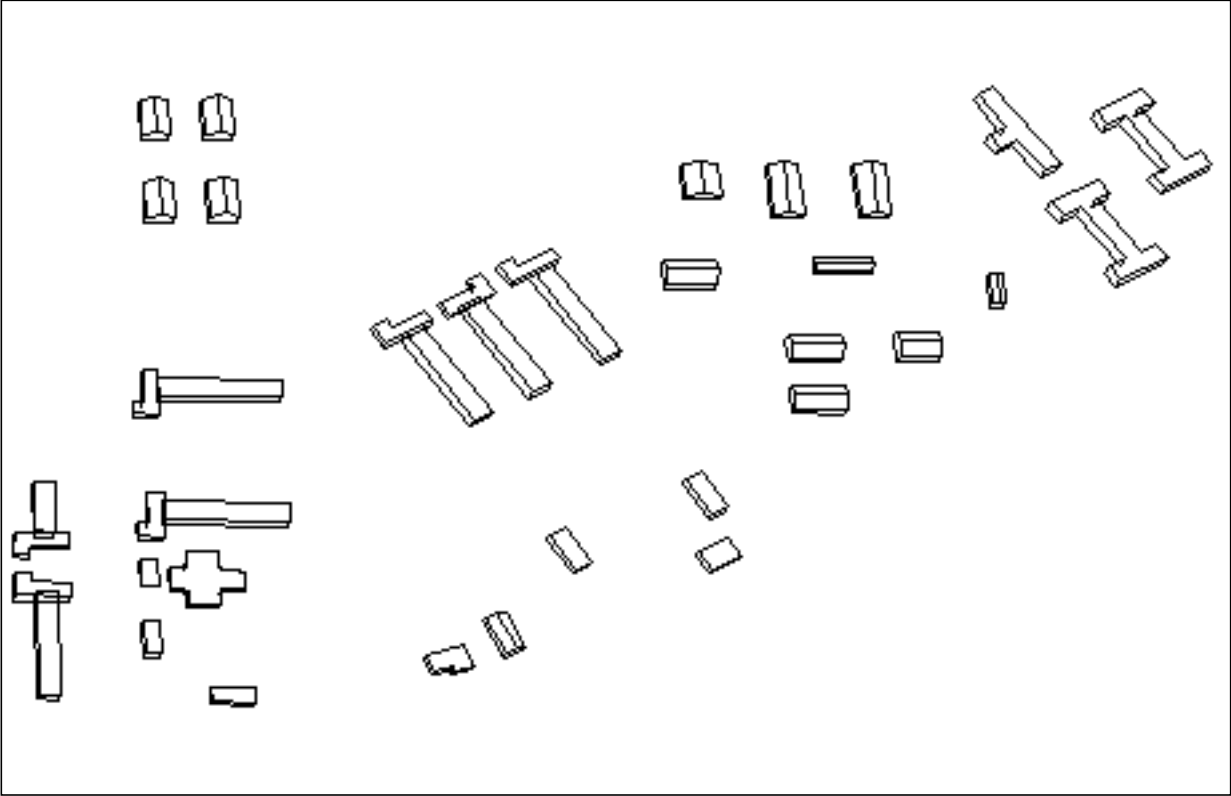


Figure 4. Portion of a Site Model

3 Site Model Validation

The purpose of model validation is to verify whether the objects in the site model are present in the new image in the same form or whether they should be examined in more detail for evidence of significant changes. The previous step of model and image registration helps provide a correspondence between the two at line segment level. For model validation, we combine evidence from a variety of object features such as lines and junctions. We also look for 3-D evidence; in a single image, this is done by evaluating expected shadows cast on the ground; in multiple images, this could come from feature correspondence information. Note that not all the features of the model may be visible in the image, some will be missing due to self and mutual occlusion. These occlusions can, however, be predicted from the viewing geometry and accounted for. There will also be missing evidence due to difficulties of feature extraction in images: low contrast edges may not be detected and line segments fragmented due to surface and ground texture.

We evaluate five kinds of evidence: edge visibility, edge presence, edge coverage, junction presence and shadow presence (these terms are explained in section 3.3). The confidence values derived take into account only visible elements from the particular viewpoint of the image after accounting for self and mutual occlusion. Before confidence values are calculated, the system deals

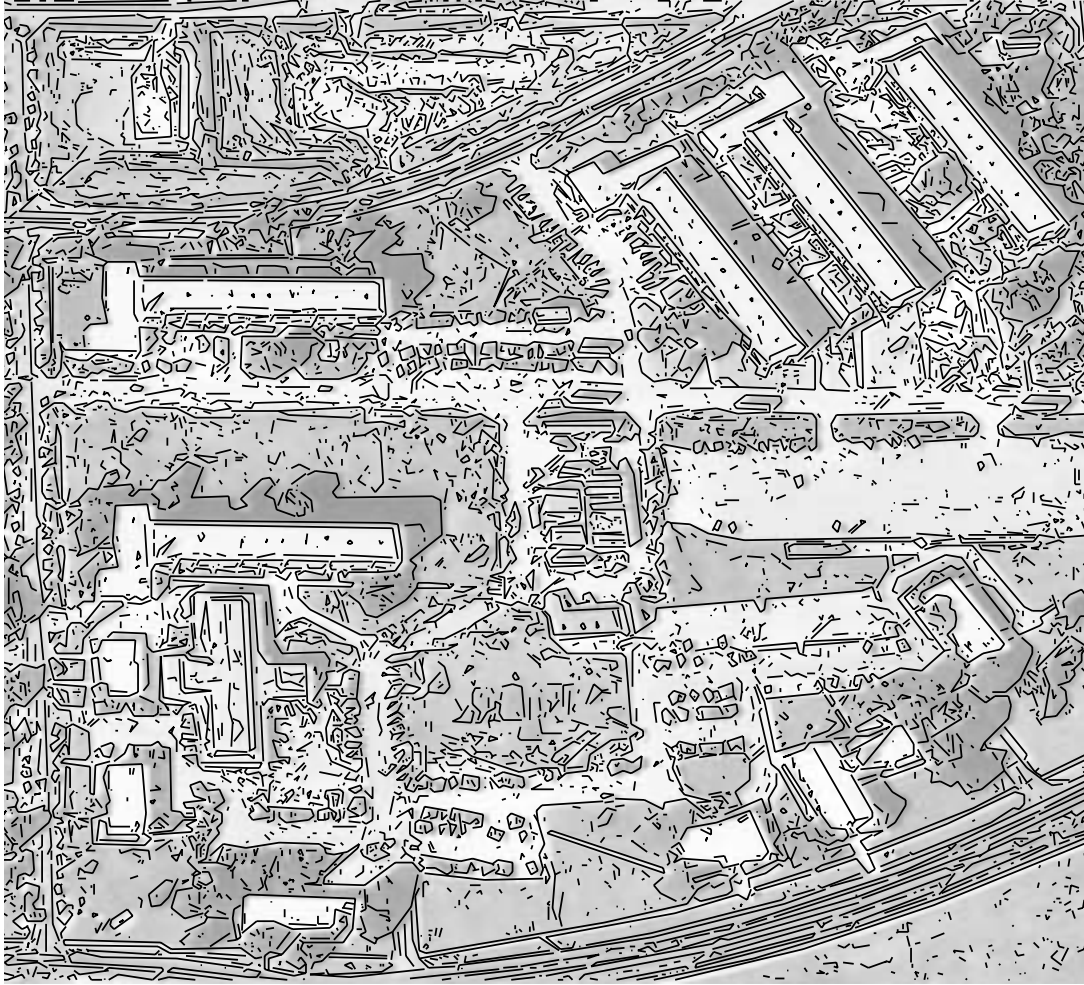


Figure 5. Line segments extracted from a portion of an image from Fort Hood, Texas

with a number of problems and ambiguities that need to be resolved, all inherent to any matching process, namely missing and multiple matches and coincidental alignments. We discuss these next.

3.1 Missing Features

To validate a model accurately we need to study the source of missing model-to-image correspondences. Some missing image features will be due to viewing conditions such as self-occlusion, occlusion by other objects, self shadows and shadows cast by nearby objects. These, however, can be predicted and explained from the site model itself. Missing correspondences may be due to over- or under-modeling of objects (Figure 8 and Figure 9) and are more difficult to predict from the model. The confidence associated with over- or undermodeled objects may thus be underestimated or difficult to calculate.

Over-modeling is due to the use of modeling primitives that introduce elements that do not correspond to actual physical elements or boundaries. Figure 8 shows a building that has been modeled by two rectangle parallelepipeds. The thick lines represent portions of the elements on the building model that do not correspond to physical boundaries. These can not be matched and the missing correspondences result in lower confidence.

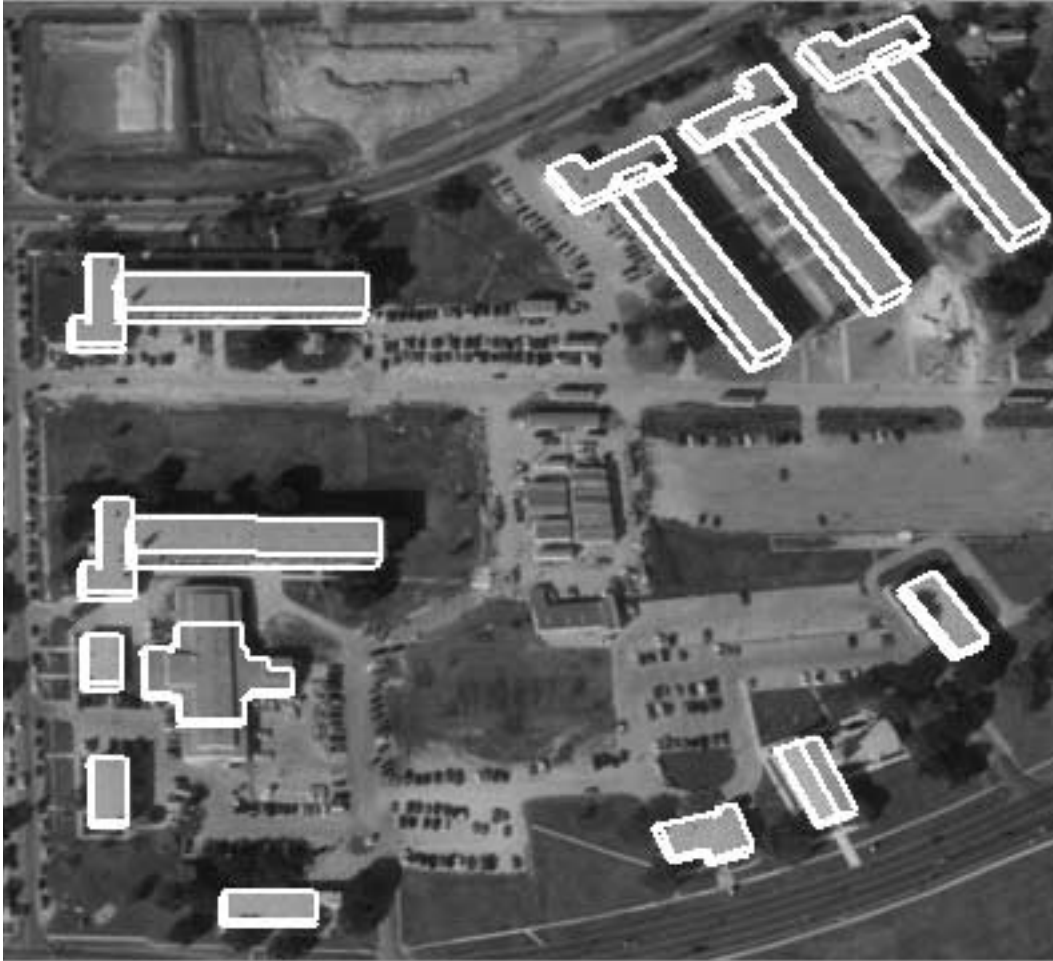


Figure 7. Site model registered with image

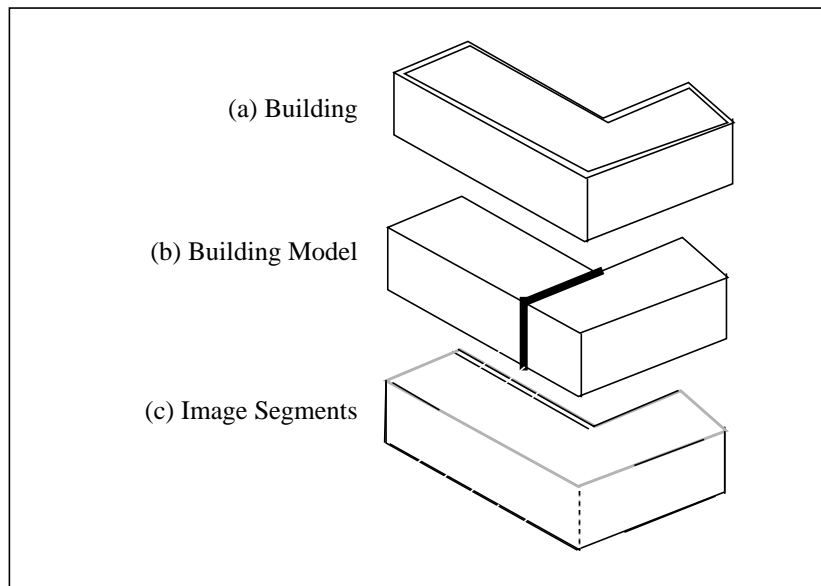


Figure 8. Missing match due to over-modeling.

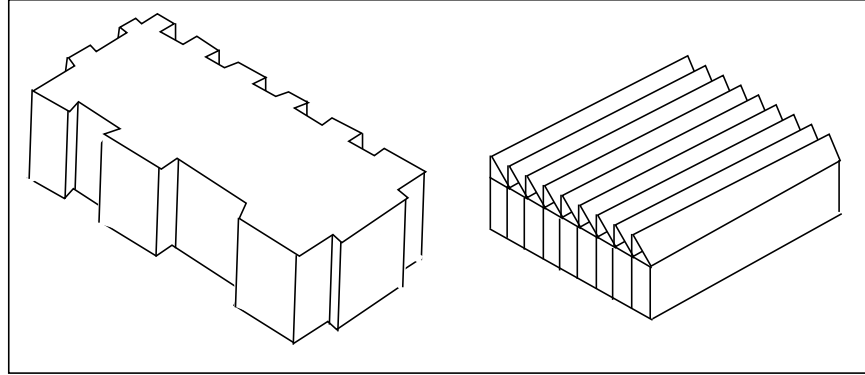


Figure 9. Some buildings may be under-modeled

Figure 9 shows two buildings that are likely to be under-modeled (i.e. modeled by simpler shapes) due to their complexity. These require additional search strategies designed to look for additional and possibly fragmented evidence, such as a large number of vertical or horizontal edge elements. Our system is currently not capable of determining these conditions, and thus the confidence values may be underestimated. It is assumed that some of these conditions may require annotations in the site model to help the system process these appropriately.

3.2 Ambiguities in Matching

There are several ambiguities inherent to the matching process that need to be resolved during validation. Our system currently deals with two of these: multiple matches, and coincidental alignments due to viewpoint, illumination direction, or due to adjacent structures.

3.2.1 Multiple Matches

The model-to-image matcher in the system corresponds each model element with one or more image elements. This is necessary to deal with expected fragmentation in the image elements. Fragmentation is due to inadequacies in the feature extraction process and due to actual image content, such as occluding trees, road boundaries and shadows. This may result in one-to-many correspondences (Figure 10) possibly involving more than one object. If a model segment matches multiple colinear image segments, all the image segments are considered to represent image support. If a model segments matches multiple parallel image segments, the overlap among these is considered to represent image support.

3.2.2 Coincidental Alignments.

Some multiple matches are due to coincidental alignments of buildings elements with elements of other adjacent or nearby structures (Figure 11). Some of these include roads, walkways, lawns, shadows, loading ramps, large vehicles, sidewalks, etc. Aligned elements may be thus detected in the image as longer line segments than expected, or predicted by the corresponding model elements. Since model segments matching longer image segments indicate a possibility of change (extension of dimension), we examine these cases to handle possible alignment in two cases: by examining nearby shadows with knowledge of the direction of illumination, and by examining adjacent structures. Coincidental alignments occur often in denser urban environments and are due to nearby and adjacent structures. We explicitly look for these to declare alignment or to declare change in dimensions. An example of some ambiguities and alignments is shown in Figure 12. If

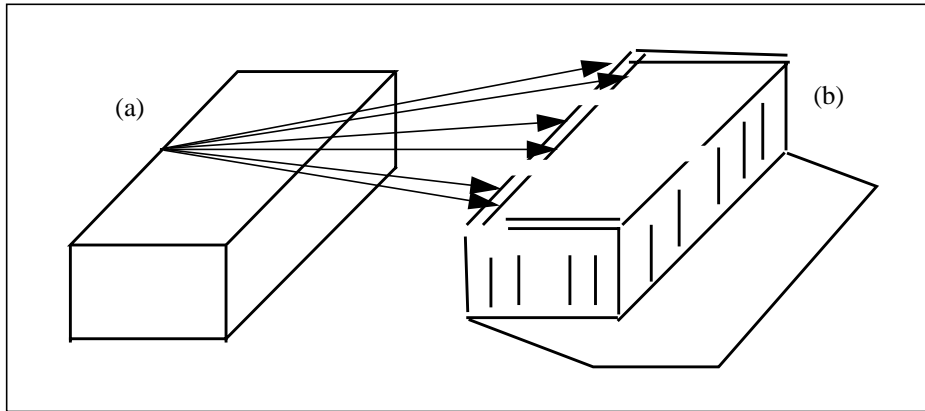


Figure 10. One-to-many correspondences.

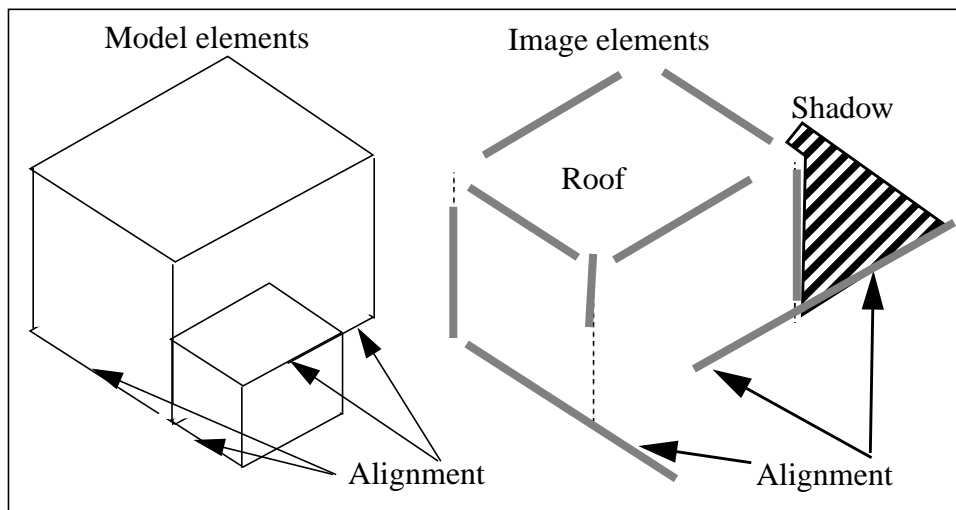


Figure 11. Coincidental alignments.

alignment is discovered, the extended portion, whether belonging to an adjacent structure or to a shadow boundary, is ignored.

3.3 Validation Confidence

We evaluate five kinds of evidence: edge visibility, edge presence, edge coverage, junction presence and shadow presence (these terms are explained below). Each evidence provides a score between 0 and 1 and a combined confidence score is computed by a linear weighted combination of them. We have chosen this method of combination for its simplicity. It works well in our tests but there may well be more optimal ways of combining such evidence. The confidence values derived take into account only visible elements from the particular viewpoint of the image after accounting for self and mutual occlusion. (Figure 13 and Figure 14):

Let x be a model object defined by a set of vertices and a set of edges. For each object, x , we calculate a confidence value $C(x)$ as a contribution of the following terms:

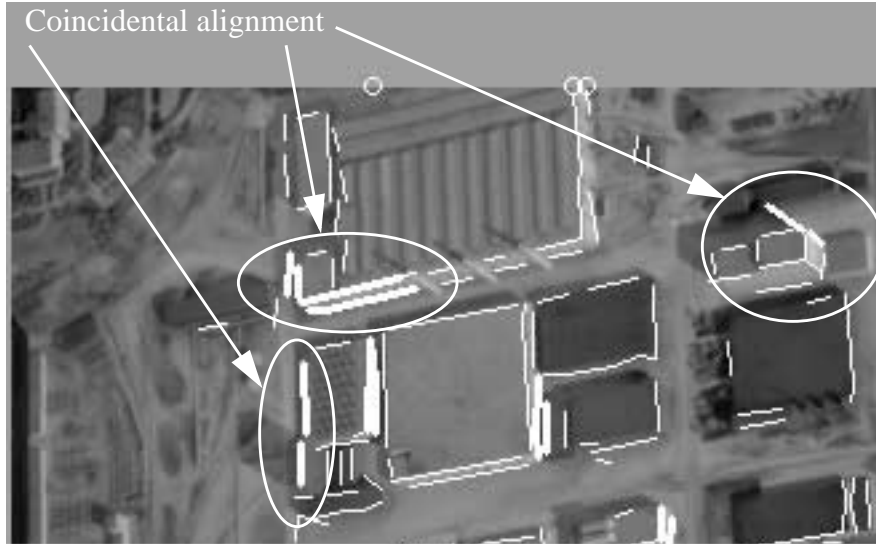


Figure 12. Example of ambiguities

Edge Visibility: $V(x)$ is given by the fraction of the model edges that are visible from the current viewpoint. Fewer visible edges result in lower confidence.

Edge Presence: $P(x)$ is defined as the fraction of the visible model edges that are matched to some image edges. In the schematic example shown in Figure 13 (a), all nine visible edges (dashed lines) have correspondences in the image (solid lines), giving a P value of 1.0. An object that is only 50% visible but that has the visible 50% corresponded to image edges has a P value of 1.0 also. P is calculated separately for **roof** elements, **vertical** wall elements and **base** wall elements which are given different weights (roof evidence is considered the most reliable, the wall base the least reliable).

Edge Coverage: $E(x)$ is defined as fraction of the lengths of the visible model edges that is actually covered by some image segments. Figure 13 (a) shows an object where all the model edges (dashed) have some, but small coverage; this object has good presence but poor coverage. Figure 13 (b) shows the opposite; a few model edges have good image edge support so the coverage is good but presence is not. $E(x)$ is also calculated separately for roof and wall elements. $E(x)$ is penalized by fragmented support. Fragmentation, $F(x)$, is defined as the ratio of the number of image segments to model to the number of model segments.

Junction Presence: $J(x)$ is defined as the ratio of the number of image L-junctions at locations predicted by the model (Figure 14) to the number of visible model vertices. Image junctions are extracted from the image from the line segments used for matching.

Shadow Presence: $S(x)$, is defined as the ratio of the number of potential shadow boundaries and junctions extracted from the image over number of visible shadow elements (boundaries and junctions) derived from the model (Figure 14). The image segments are labelled as potential shadow segments by noting the consistency of the “dark” side of the segment with respect to the direction of illumination. Segments oriented parallel to the projection of the direction of illumination correspond to possible shadow lines cast by vertical edges. The L-junctions formed (allowing for gaps) by potential shadow lines are labeled potential shadow junctions. Details on shadow evidence extraction may be found in [12].

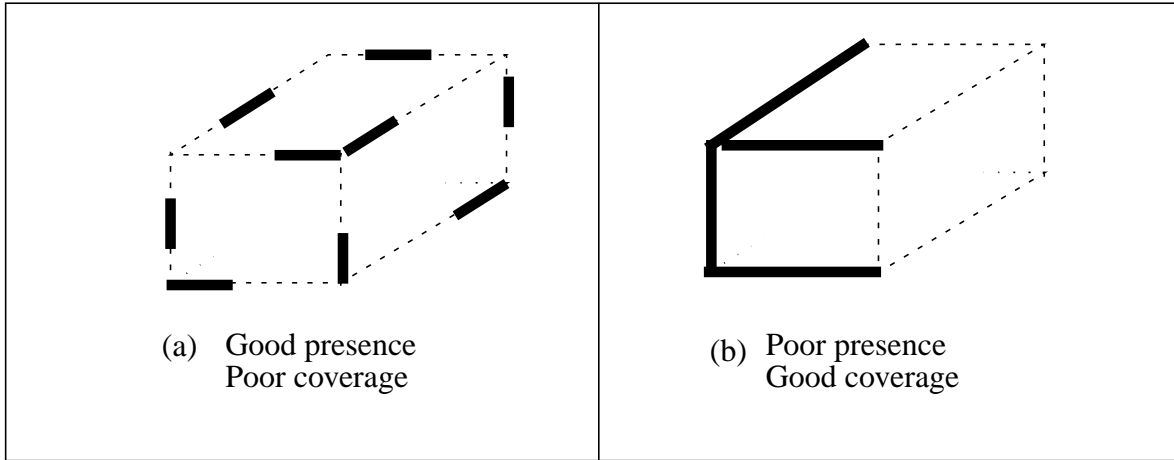


Figure 13. Presence and Coverage.

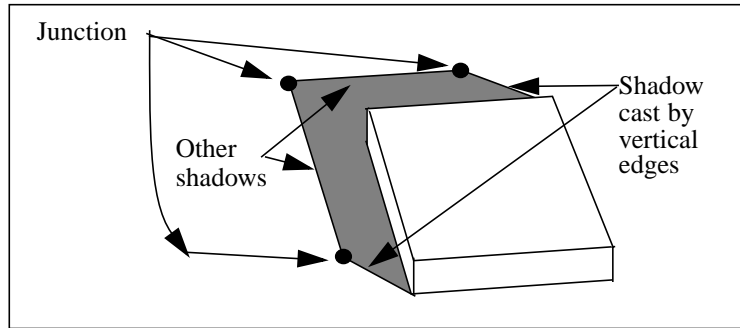


Figure 14. Shadows cast by a rectangular

A combined confidence value $C(x)$ is calculated by a linear weighted sum of the above described pieces of evidence, with $w_p = 7$ (roof), 5 (wall), and 3 (base), $w_v = 1$, $w_s = 3$, $w_j = 3$ and $w_m = w_p$:

$$C(x) = \frac{1}{\sum_i w_i} \cdot [w_p \cdot P(x) + w_v \cdot V(x) + w_s \cdot S(x) + w_j \cdot J(x) + w_m \cdot \frac{E(x)}{F(x)}]$$

High confidence values indicate good image support while low values denote low image support. Low values may signify change as lack of image support may be due to missing buildings, or buildings that have undergone significant change with respect to their current model. However, model buildings that have strong image support, may have changed also as additions to structures, such as a new wings, may not affect significantly the appearance of the previously modeled portions.

Figure 15 shows the results of the validation step applied to the image shown in Figure 7 earlier. The labels indicate the validation confidence level as a function of C values: high (**H**) for $C \geq 0.5$; medium (**M**) for $0.4 \leq C < 0.5$ and low (**L**) for $C < 0.4$. Note that a building indicated as having low (**L**) confidence has actually drastically changed (or was grossly mis-modeled) whereas the ones marked high (**H**) are in fact, unchanged. Medium level (**M**) typically denote buildings with moderate or “acceptable” image support. These assignments are arbitrary however, and would have to be set as a function of the task at hand. Some applications may require detailed explanations of possible change that require higher discrimination. Here we show only three for simplicity.

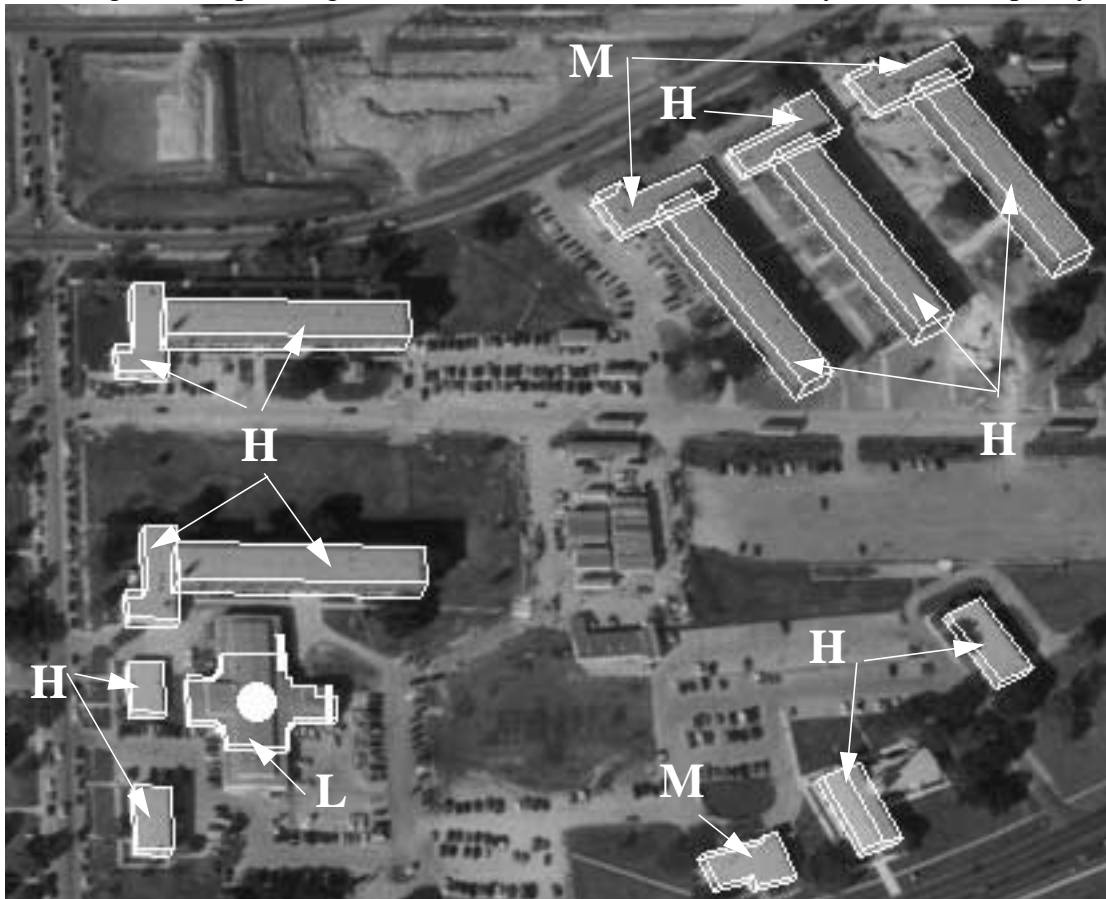


Figure 15. Validation result and labeled confidence levels.

4 Structural Change Detection

The validation step (validation) makes available information that is used to start analyses to determine structural changes. Two cues are used to investigate structural changes:

Validation Confidence values: These values reflect image support for a model object. Although low support may be due to poor image quality, lack of contrast, occlusion, or viewpoint, they can signify missing structures, substantially altered structures or incorrect modeling. Medium level support denote “acceptable” indication of presence with reduced support due to poor image quality, lack of contrast and other image dependent characteristics. High values clearly denote strong presence and image support, at least for the modeled portions.

Extra Image Elements: Model elements that correspond to image elements having greater extent (size) than that of the model elements provide preliminary indication of possible changes in the dimensions of the object.

The above cues are used to further analyze whether one of the following three classes of structural change has occurred: missing (or mis-placed) buildings, dimensional changes, and new buildings. The methods to infer these changes are described next.

4.1 Missing Buildings

Model buildings having very low confidence values denote poor image support. The possible causes for this condition are that either the model is incorrect, the structure is heavily occluded or that the building has been removed or destroyed (assuming that images are of sufficient quality), or that its position is grossly incorrect. A low confidence is sufficient to report a missing building, if additional images were available, they could be examined for confirmation. Figure 16 shows the result in a small window containing two modeled buildings. The model building on the right was added to the model by hand to test this condition. It is reported as having low confidence correctly as evidence for its presence can not be found in the image. Other examples of this condition are shown later in section 6.

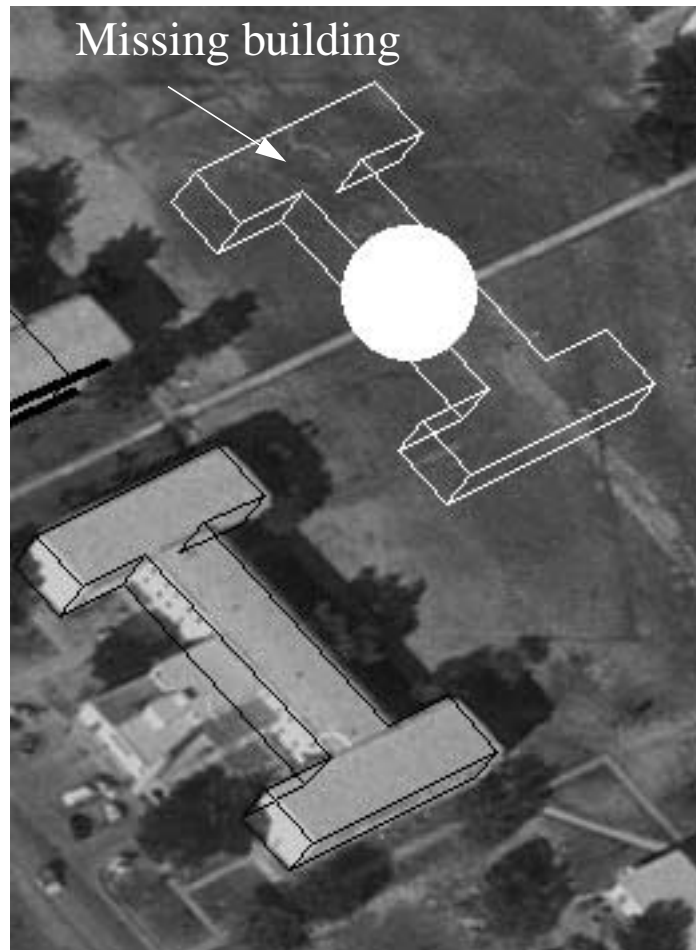


Figure 16. Missing building (white outlines)

4.2 Validated Buildings

Model buildings having moderate (**M**) to high (**H**) levels of support are considered validated. That is, their presence in the image is verified. The buildings labeled (**M**) typically require verification in another image to increase, if possible, their confidence level. An example is shown in Figure 17. The L-shaped building in the small window corresponds to the L-shapes building, labeled **M** on the top middle of the image shown in Figure 15. From this viewpoint the confidence value of this building increased to 0.5661, thus becoming validated with high confidence.

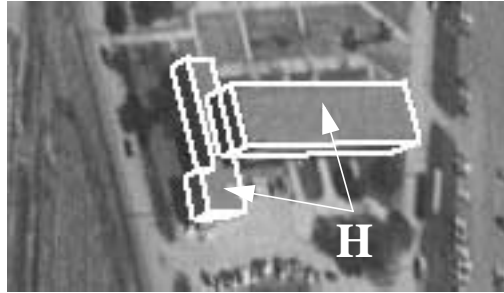


Figure 17. Validation of L-shaped building from another viewpoint.

4.3 Dimensional Changes to Modeled Structures

In some cases, regardless of confidence, the system is able to cue possible structural changes based on the fact that model edges are matched to **longer** image edges. Those cases where these conditions arise due to accidental alignments have been handled earlier, as explained in section 3.2. The remaining cases therefore represent cues for possible change. We can apply a building finding tool at this stage to confirm a change (and to derive a description of the change) as shown later in section 5, however, this relies on the building finder's ability to find new buildings. A less stringent criterion may be to only search for some additional evidence such as the extra edges casting a shadow and/or having other evidence of being above ground (perhaps by using multiple images). We have not implemented such partial analysis though components of it are available in the building finding tool.

Figure 18 (a) and (b) show two examples of evidence that support cueing dimensional changes. The matching and fine registration step correctly registers the modified models to the structures in the image. The thick white gray lines are the extended image segments that matched the corresponding model edges thus triggering the cue.

4.4 New Buildings

One important type of site change is the introduction of new structures. For such changes, the previous model is less useful but can still be relied on to provide some context. Such context can consist of areas of interest and characteristics of existing buildings (to check if buildings similar to existing ones have been constructed). Our system only uses the site model to mask out the modeled areas and a building finding tool is applied to all other areas, or all other areas containing large number of unmatched image features such as corners. The camera models and terrain models associated with the site are used to derive viewpoint and illumination parameters automatically. An

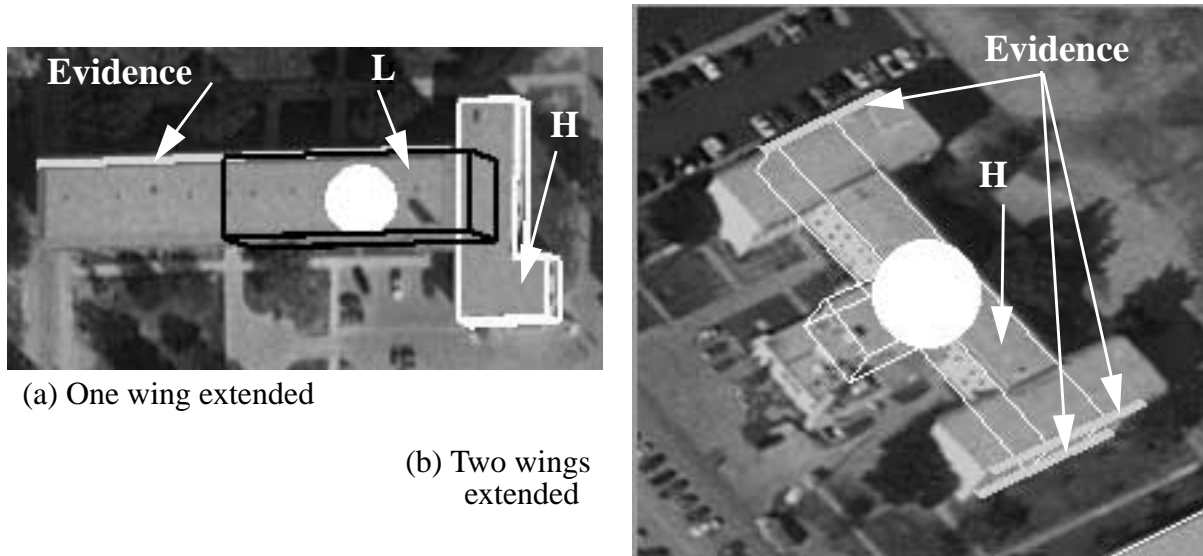


Figure 18. Actual change in dimensions.

example of this task is the next section (Figure 25). We have experimented with focus of attention mechanisms to select the areas where automated detection should be applied.

5 Model Updating

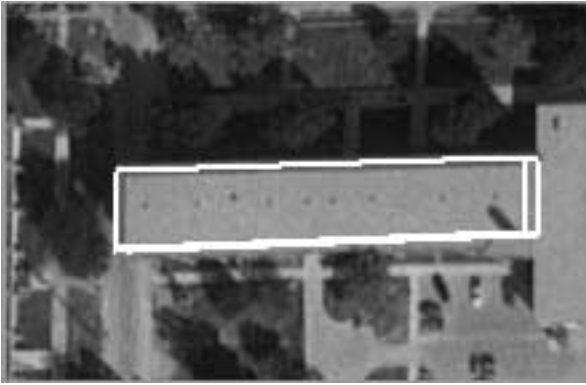
The next task is to make a model for the detected structural changes and to incorporate it in the site model. We describe two situations: where changes are made to existing structures and where new structures are detected.

5.1 Modeled Buildings

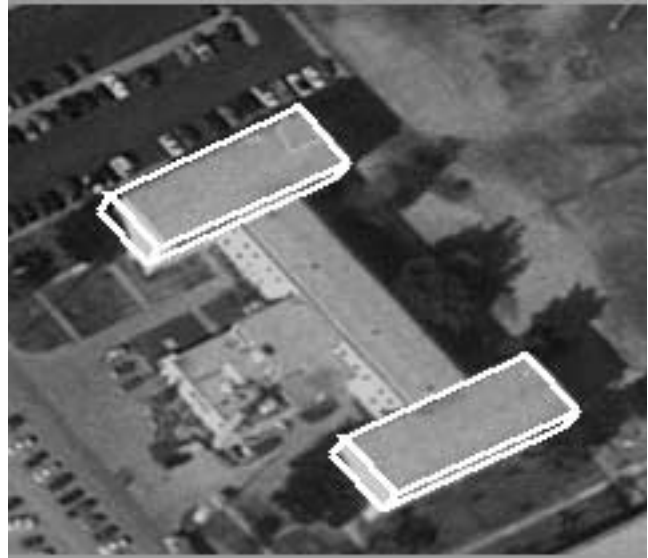
Changes in the dimensions of modeled structures that have been cued by the previous step need to be analyzed further, possibly using more than one view of the scene, if available. We use a monocular building detection system [12] to return the highest rated building hypothesis that can be formed in the location of the cued change. These are shown in Figure 19 for the two examples shown earlier in Figure 18. In this examples the image locations and sizes were selected manually before application of the automatic detection system. The 3-D models are however derived automatically (using shadow and wall evidence for 3-D inference).

5.2 New Buildings

Detection of new buildings is a more difficult task as the site model is less helpful. Our method consists of simply applying a building finding tool to areas of interest and reporting the results that are of sufficiently high confidence. Figure 20 shows the result of applying a monocular building detection system [12] to look for change in the form of new buildings (shown with white outlines). The areas modeled are ignored by the system. Typically the system would be instructed to locate new buildings in designated areas that are of interest, such as functional areas. The three buildings shown in white outlines are detected automatically and become suggested candidates to be added to the site model.



(a) 3-D model generated by building detector for the extended wing.



(b) 3-D models suggested for changed wings.

Figure 19. Suggested updating of cued changes

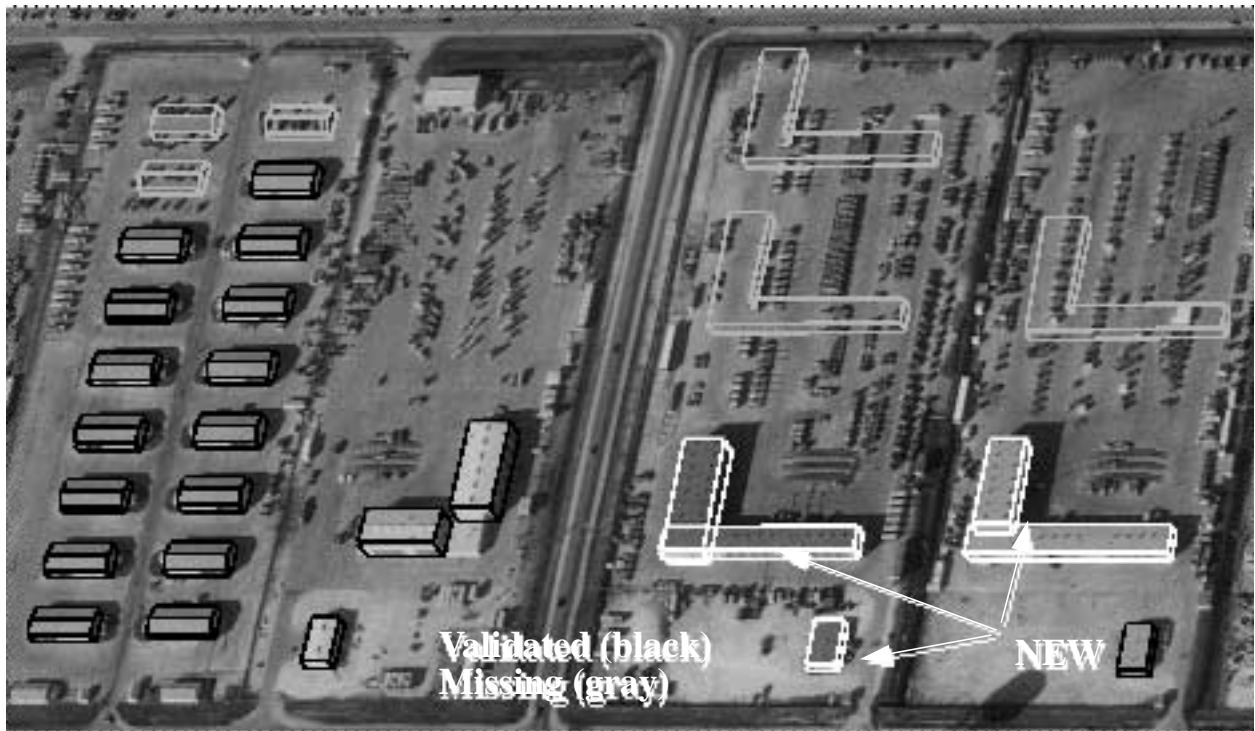


Figure 20. Model updating. New buildings are detected automatically

6 More Validation and Change Cueing Results

We show some additional examples and results in this section. Figure 21 shows an image of the Fort Hood site with the site model overlaid on it. The image size is 7775x7720 pixels, and the 3-D site model contains 79 objects representing building structures. In this example, the image and the

model are registered within a small translation. The system therefore is applied to each model object separately using small image windows. Processing time is about 15 seconds per structure on a Sun sparc-10 workstation, running under the RCDE (Radius Common Development Environment.) The results are summarized in table 1. It shows the number of building objects visible in the image and the distribution of validation confidence values (the label codes in the table are in reference to the graphical results shown below in Figure 22 through Figure 25. **H**, **M** and **L** denote high, medium and low validation confidence respectively). The confidence values are dependent on the image content and may not necessarily reflect structural changes but in this example at least, there is a high correlation between confidence level and the number of buildings changed, not changed or missing in the rest of the table. All matching ambiguities, with one exception, shown in Figure 22, are correctly handled. This case involves an alignment with a ground feature not present in the model, a situation, not currently handled by the system. Of the 54 non-changed buildings, one is cued incorrectly to have changed. This building is shown on the left of Figure 22, and is the same one for which the unresolved coincidental alignment appears to indicate that the building has been extended towards the left. This situation is however likely to be corrected by confirmation of the change using another view of the building.

Fourteen buildings that are actually present in the image had changes. Thirteen of these are found to be changed correctly: Three in Figure 22, five in Figure 23, three in Figure 24 and two in Figure 25. These are shown in the figures with a circle on top and thick white lines cueing evidence of dimensional changes. The remaining changed building is the L-shaped building on the right of figure23. The building has two wings, both of which have changed. Only the left wing is detected to have changed. This situation is likely to be corrected by reconciling the output from more than one view.

Buildings that change considerably or are missing have poor image support, resulting in low validation confidence (the buildings labeled **L** in Figure 22 through Figure 25). There are 12 of these, 11 of which were added by hand to test the “missing building” detection capability. The remaining one, represents a significantly changed building (The cross-shaped building in Figure 24). All these are labeled correctly as changed or missing.

Table 1: Summary of Results

Image (fhov927)	Visible Buildings	Validation Confidence			Non-changed Buildings			Changed Buildings			Missing Buildings		
		High (H)	Medium (M)	Low (L)	Number of buildings	Reported non-changed	Reported changed	Number of buildings	Reported changed	Reported non-changed	Number of buildings	Reported missing	Validated
No.	79	54	13	12	54	53	1	14	13	1	11	11	0
%	100	68.3	16.4	15.1	100	98.1	1.9	100	92.8	7.3	100	100	0.0

7 Choice of Parameters

Our system uses several parameters in its decision making processes at various levels. Choice of these parameters, of course, determines the quality of the results that are obtained even though we have attempted to make the system not very sensitive to them. Ideally, the parameter values should be based on a mathematical analysis of the algorithms and be a function of the parameters of the input images and of any known parameters of the site. Unfortunately, such an analysis is difficult due to the complexities of the algorithms and estimating appropriate image parameters. In our system, we make decision parameters a function of the image parameters that are supplied with the images, such as the resolution. Additional refinements may be possible by using context of the site. We have set the parameters by an informal analysis of the process and testing with available sets of data. All of our examples use the same parameter values.

8 Conclusion

We have shown some results and capabilities of our system for detecting and describing structural changes. It has been tested on real images (though with simulated changes to the model) and seems to be quite effective at finding significant changes in rather complex images. It is able to find missing (or misplaced) buildings, buildings with changed (or incorrectly modeled) dimensions and new buildings (or previously unmodeled buildings). This system relies on the use of a single image to find changes. We anticipate that its performance would be significantly enhanced by use of multiple images as they would provide independent evidence of changes and also allow the reasoning to proceed more directly in 3-D space. Several multiple image building finders are becoming available [13, 14, 15] and could be easily incorporated in our method. Our system has been ported to an industrial laboratory for possible use in current applications.



Figure 21. Portion of an image from Fort Hood, Texas.

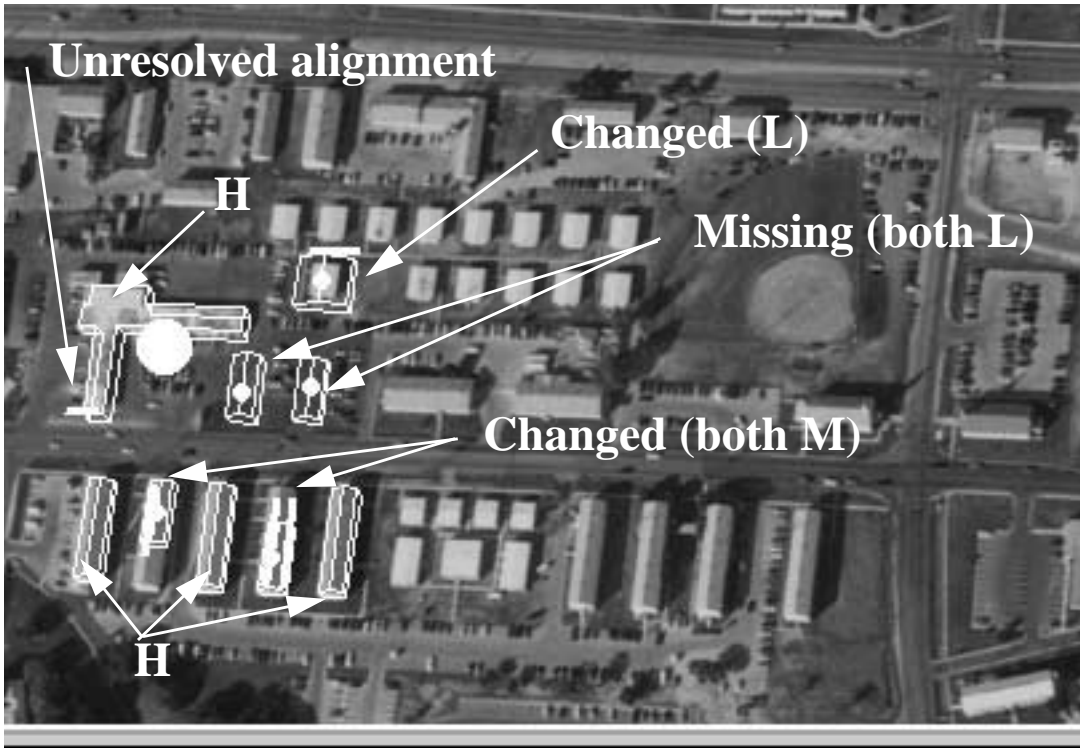


Figure 22. Validation and change detection result for area A of Fort Hood



Figure 23. Validation and change detection results for AREA B of Fort Hood example



Figure 24. Validation and change detection result for AREA (C of Fort Hood, Texas

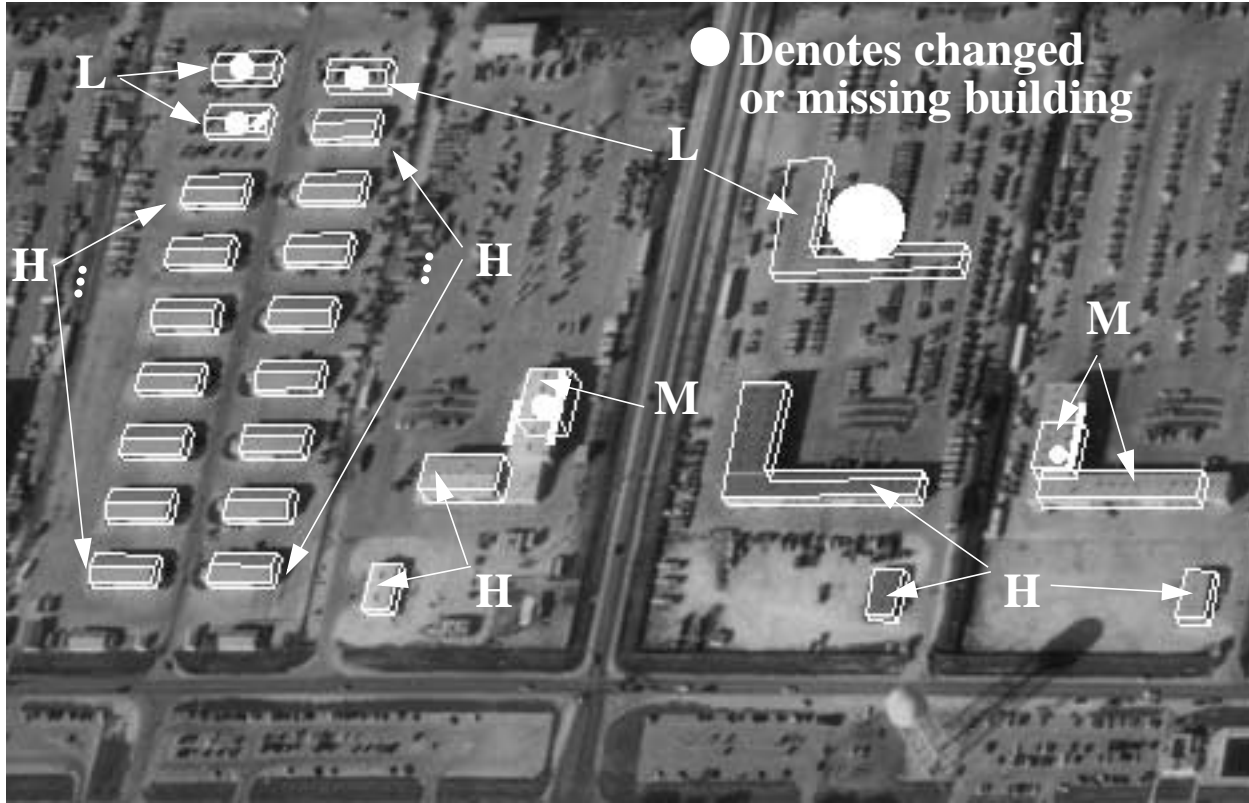


Figure 25. Validation and change detection result for AREC D of Fort Hood.

References

- [1] Lillestrand, R. L., "Techniques for Change Detection," *IEEE Transactions on Computers*, 7 (1972), pp. 654-659.
- [2] Khalaj, B., Aghajan H. and Kailath T., "Automated Direct Patterned Wafer Inspection," *Proceedings of the IEEE Workshop on Applications of Computer Vision*, Palm Springs, California (1992) , pp. 266-273.
- [3] Gerson, G. and Wood S., "The RADIUS Testbed System," in: Firschein, O and Strat, T, ed, RADIUS: Image Understanding for Imagery Intelligence, (Morgan Kaufmann Publishers, San Francisco, 1997), pp. 5-12.
- [4] Huttenlocher, D. and Ullmann S., "Recognizing Solid Objects by Alignment with an Image," *International Journal of Computer Vision*, Vol. 5, No.2., November, 1990, pp. 195-212.
- [5] Chellappa, R., Zhang X., Burlina P., Lin, C., Zheng Q., Davis L. and Rosenfeld A., "An Integrated System for Site Model Supported Monitoring of Transportation Activities in Aerial Images," *Proceedings of the DARPA Image Understanding Workshop*, Palm Springs, California, (Morgan Kaufman Publisher, San Francisco 1996), pp. 275-304.
- [6] Strat, T., Quam L., Mundy J., Welty, R., Bremner, W., Horwedel, M., Hackett, D. and Hoogs, A, "The RADIUS Common Development Environment," *Proceedings of the DARPA Image Understanding Workshop*, San Diego, California, (Morgan Kaufman Publisher, San Francisco, 1992), pp. 215-226.
- [7] Fua, P., "Cartographic Applications of Model-Based Optimization," *Proceedings of the Image Understanding Workshop*, Palm Springs, California, (Morgan Kaufman Publisher, San Francisco, 1996), pp. 409-419.
- [8] Huertas, A. and Nevatia R., "Detecting Changes in Aerial Views of Man-Made Objects," *Proceedings, IEEE Sixth International Conference on Computer Vision*, Bombay, India, (January, 1998), pp. 73-80.
- [9] Médioni, G., Huertas, A., and Wilson, M., "Automatic Registration of Color Separation Films," *Machine Vision and Applications*, 4 (Springer-Verlag, New York, 1991), pp. 33-51.
- [10] Huertas, A., Bejanin, M. and Nevatia, R., "Model Registration and Validation," in: Gruen, A., Kuebler, O. and Agouris. P., ed, Automatic Extraction of Man-Made Objects from Aerial and Space Images, (Birkhauser Verlag, Basel 1995), pp. 33-42.
- [11] Hough, P., "Methods and Means for Recognizing Complex Pattern,s" U.S. Patent 3069654, December 18, (1962).
- [12] Lin, C., Huertas, A. and Nevatia, R., "Detecting Buildings from Monocular Images," in: Gruen, A., Kuebler, O. and Agouris, P., ed, Automatic Extraction of Man-Made Objects from Aerial and Space Images, (Birkhauser Verlag, Basel 1995), pp. 125-134.
- [13] Noronha, S. and Nevatia, R., "Detection and Description of Buildings from Multiple Aerial Images," *Proceedings of the DARPA Image Understanding Workshop*, Palm Springs, California, (Morgan Kaufman Publisher, San Francisco, 1996), pp. 469-478
- [14] Jaynes, C., Stolle, F. and Collins, R., "Task Driven Perceptual Organization for Extraction of Rooftop Polygons," *Proceedings of the ARPA Image Understanding Workshop*, Monterey, California (Morgan Kaufmann Publishers, San Francisco, 1994), pp. 359-365.
- [15] Collins, R., Hanson, A, Riseman, E. and Schultz, H., "Automatic Extraction of Buildings from Aerial Images," in: Gruen, A., Kuebler, O. and Agouris, P., ed, Automatic Extraction of Man-Made Objects from Aerial and Space Images, (Birkhauser Verlag, Basel 1995), pp. 169-178.

