

User Assisted Modeling of Buildings from Aerial Images*

J. Li, R. Nevatia and S. Nornoha

Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, California 90089
nevatia@iris.usc.edu

Abstract

An approach that allows a user to assist an automatic system in modeling buildings is described. The approach is designed to be efficient in user time and effort while preserving the quality of the models created. Currently our system is able to handle the rectangular buildings with flat roof or symmetric gabled roof. Models can be created by only one or two clicks in many cases. Efficient editing of automatically derived models is also possible.

1 Introduction and Overview

Generating 3-D models from a set of images is a common task for computer vision. In spite of substantial research in this area, the performance of machine algorithms remains significantly below that of humans. In this paper, we explore an approach to bridge this gap by allowing a human “in the loop” but by requiring only simple interactions from the user to generate accurate models efficiently. We illustrate this approach for the task of building modeling from aerial images which is a difficult and important task.

Significant progress has been made in recent years in the goal of extracting models of buildings from aerial images by completely automatic systems [1] [2] but the results are not completely accurate. Completely manual systems require an unacceptable amount of effort from a human modeler, both in terms of time and cost. We describe an approach that attempts to provide user assists to an automatic system in a way that the user effort is diminished significantly while the quality of the results is still preserved.

Several approaches to user assisted modeling are possible. The conventional approach is to provide a set of generic models which are then fit to the image data by changing model and viewing parameters [7]. In this approach, the system provides geometric computations but substantial time and effort are required from the user. Newer approaches have attempted to combine user input with varying amounts of automatic processing. In [3], the authors suggest

providing just an approximate building location to extract a building. In [4], other interactive tools are described including methods for replicating to model buildings that are identical or very similar to others. In [5], an automatic system constructs topological relations among 3-D roof points collected by a user for each roof; this system can work with several types of complex roofs.

In our approach, basic modeling tasks are still performed by an automatic system but this system receives simple, but critical, assists from a user. The assisted system’s capabilities are limited by those of the underlying system. In our case, the shapes of the buildings are restricted to be rectilinear; the roofs may be either flat or symmetric gables.

The underlying automatic system is the multi-view building detecting system described in [6]. The basic steps of this system consist of forming parallelogram hypotheses (to represent rectangular parts of roofs) in one image (but by using information about matched lines) and inferring 3-D shapes from them. 3-D analysis also provides verification of the hypotheses by determining if sufficient evidence for their presence exists. A user can assist this system in the process of hypothesis formation as well as in making corrections to the resulting 3-D models.

A user interaction typically consists of the user pointing to a point or line feature; the pointing need not be precise as precise features are automatically selected by the system. We will call one such interaction a “click”. The system requires two (or more) views of a scene with associated camera geometry; however, all user interactions take place in one view only. Other views can be displayed but the user is not asked to view the images stereoscopically. We believe that confining interactions to one view can significantly reduce the effort required by the user.

The following system description is divided in two components. In the first situation a building has not been detected and needs to be added. In the second, a building has been detected partially and requires editing.

2 Adding a Building

User assistance in adding a new building to the model consists of approximately indicating some numbers of cor-

* This research was supported in part by the Defence Advanced Research Projects Agency of the Department of Defence under contract # DACA76-97-K-0001 and monitored by the Topographic Engineering Center of the U.S. Army.

ners of its roofs. The automatic system constructs and displays a 3-D model after each click (though in some cases, no models may result). Each subsequent click refines the models. For each rectangular component of a rectilinear building, up to three clicks may be required for flat roofed buildings and up to four clicks for gabled roofs. The resulting models can then also be edited by the methods described in section 3.

After each click, indicating the approximate position of a corner, the system finds nearby corners constructed from extracted matched line segments. The sides of a single corner can suffice to trigger some parallelogram hypotheses. Multiple hypotheses are possible at each step; selection among them is made by searching for matching evidence in all the available views. The matching process also creates 3-D models from the initial 2-D roof hypotheses.

The operational and computational processes for the flat and the gabled roofs are quite similar, however, for simplicity, we describe each of them separately below.

2.1 Flat Roof Buildings

A maximum of three corner clicks is required to specify a rectangular flat roof component.

Figure 1 depicts the situation after the first user click. The system locates all junctions near the click and reports failure if none is found. For each junction found, the system attempts to construct a parallelogram. The parallelogram is formed by first examining the stored information and looking for a U-structure that uses the junction legs. If no U-structure is available, the junction legs are used to derive the parallelogram (roof hypothesis). The elements of the parallelogram are matched to elements on the other views and scores are computed as the system would during automatic operation. The system then selects one configuration and presents it to the user.

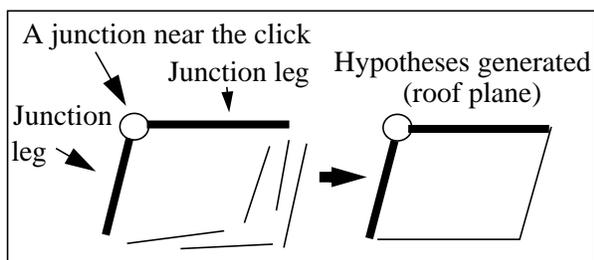


Figure 1 . First input analysis.

Figure 2 illustrates the situation after the second click. The second click is used to generate a new hypotheses in the same manner as with the first click. The hypotheses are formed that include the point from the first click however, are weighted higher.

After the third click, three points are used to form three possible parallelograms to represent roof hypotheses, as shown in Figure 3. The system calculates the 3D orientation

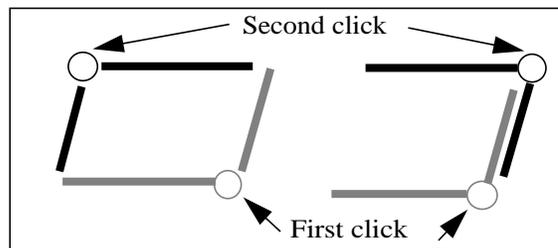


Figure 2 . Two possible configurations for a second click.

of these planes, matches the elements with elements on other views. For all possible matches, select those hypotheses with least inclination for a flat roofed building. Also the angles between the sides must be close to 90 degrees in 3D. The system computes scores as before, and selects the hypothesis with the best score.

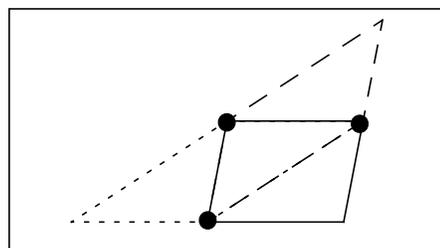


Figure 3 . Three parallelograms can be formed from three points.

To convert 2D roof hypothesis to 3D building hypothesis, we need to compute the applicable height for the roof hypothesis. This height could be computed by using matches for the linear features forming the hypotheses in two or more views. However, we have found it more robust to simply conduct a search for different height values in a range by small steps. For each hypothesized height, the 3-D model is projected to a second view and supporting evidence for it, consisting primarily of nearby and overlapping linear features is collected. Only the roof edges are considered. The height providing the best score is selected. The height of the ground is assumed to be known (or can be computed from a separate user interaction).

We show some examples of user interaction on small windows. Figure 4 shows two examples where a single click was sufficient to recover each of the two buildings, with no further editing required.

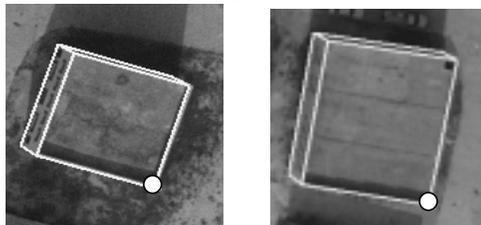


Figure 4 . Addition of two buildings by one click each.

Figure 5 shows an example where three clicks are needed. The first click results in a partial hypothesis. The second click gives a better hypothesis but is still not completely accurate. The third click results in an accurate model that requires no further editing. For each of these examples, the time required to construct a model is less than 4 seconds.

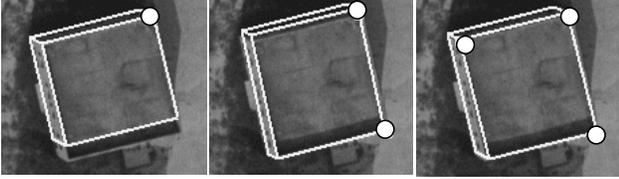


Figure 5 Addition of a building by three clicks.

2.2 Gabled Roof Buildings

At least two corner clicks but not more than four corner clicks are required to generate a rectangular symmetric gabled roof component.

First, the user clicks on or near two corners on roof spine of the desired building. The system locates all the junctions near these two clicks. If none is found for either corner, the system reports failure and prompts the user to give more clicks. Otherwise, for each junction found, the system tries to find all slanted side edges which make right angle with the spine edge. Among all these slanted side edges, the system evaluates their goodness and picks a best one. Thus we can obtain a derived junction for one side of the roof. The situation after first two clicks is shown in Figure 6.

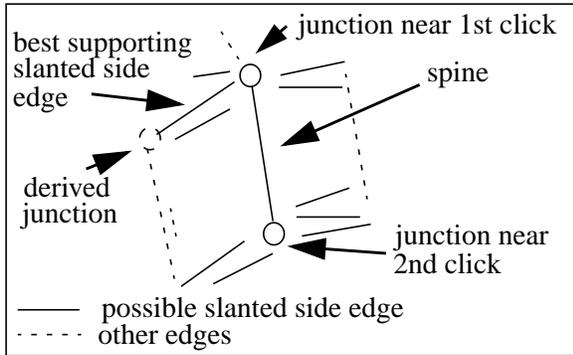


Figure 6 . The situation after first two clicks.

If the above derived junction can not be obtained or is inaccurate, the user may click on or near a corner of the roof side. With these three junctions obtained either by clicks or from computations, the system then tries to locate the symmetric corner of the current obtained side corner. Because the side height is not known yet, the system assumes a medium height for this corner and then projects it from the image coordinate to the world coordinate. Using symmetry property, the system is able to acquire the 3D coordinate of the reflected corner. Then the system projects it back to the image coordinate, locates all possible corners around, and chooses a best one as the other derived junction. Figure 7 il-

lustrates the use of symmetry property in obtaining the other derived junction.

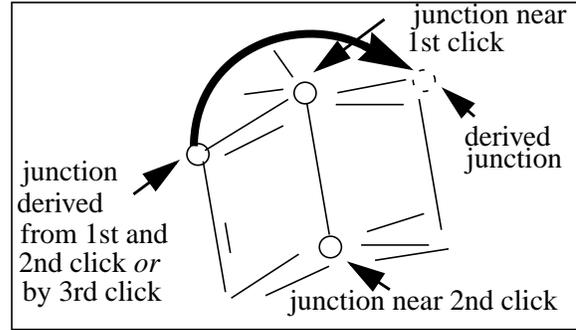


Figure 7 . Use symmetry property to try to compute the symmetric junction.

If the other derived junction is inaccurate, the user needs to provide a fourth click. With all these four junctions obtained either by clicks or from computations, the shape of the gable roof is determined and the corresponding 2D roof hypothesis can be generated, as shown in Figure 8.

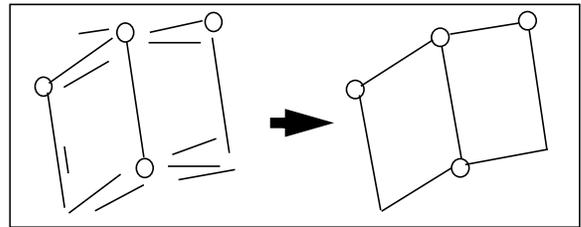


Figure 8 . 2D Gabled roof hypothesis formed.

To form a 3-D building hypothesis, we need to compute the heights of the side and the spine of the gable roof. This is done by a process similar to that for flat roofs and consists of searching in the space of allowed side and spine heights, projecting the 3D hypotheses to another view and collecting the supporting matching evidence. As this method requires a search over two parameters, a two-level matching analysis is used to speed the computations. In the first level, only the spine height is varied. For each spine segment, we compute its score as the accumulated matching spine edge evidence in another view. Only those segments with scores higher than 70 percent of the highest score are kept for continued matching analysis. In the second level, we change the side height by small steps in the allowed range. For each side segment and each remaining spine segment, we project the hypothesis into another view and computer the score as the accumulated matching roof edge evidence. The pair of side height and spine height which score best in the second level are used to convert 2D hypothesis into 3D hypothesis.

We show some examples of user interaction for the Gabled buildings. Figure 10 shows two examples where two clicks are sufficient to detect symmetric gabled buildings, with no further editing required.

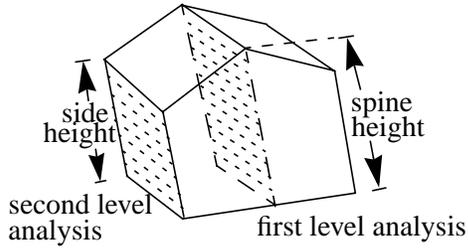


Figure 9 .Two-level analysis of height for gabled

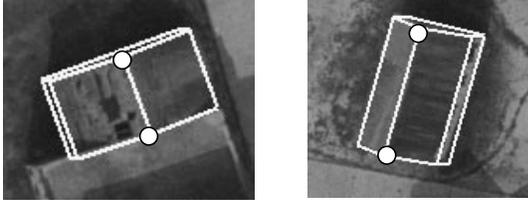


Figure 10 . Addition of gabled buildings with two clicks.

Figure 11 shows an example where a third click is needed to refine the hypothesis from the previous two clicks. The third click generates the accurate model that requires no further editing.

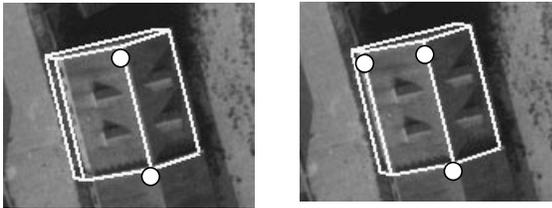


Figure 11 . Addition of a gabled building where the 3rd click is needed to refine the hypothesis

Figure 12 gives an example where four clicks are necessary. The system finds no hypothesis with first two clicks. The third click results in an inaccurate hypothesis. Thus the last click is essential to form the accurate hypothesis. For each of above gabled buildings, the time to construct a complete model is less than 6 seconds.

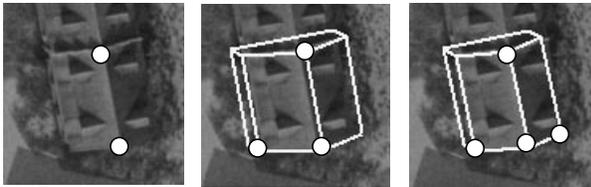


Figure 12 . Addition of a gabled building where four clicks are needed.

3 Editing a Building

This process can be used to edit a current building hypothesis, either derived automatically or by interactions in earlier stages as described above. Three available corrections are to adjust a corner, to adjust a side or to adjust

height. If more complex interactions are needed, the building should be deleted and reconstructed. Again, interactions take place in only one view and only approximate locations are needed. The user interactions are as follows:

- To adjust a corner, click on a new location for it.
- To adjust a roof side, click anywhere along the actual roof side.
- To adjust height, click anywhere along the base of the building.

For each of the steps above, the system recomputes all aspects associated with the formation of 3D hypothesis during automated operation. The actions of the automatic system for editing a building are similar to those for adding a buildings. If a corner is indicated by the user, the system finds the nearest corner in the existing hypotheses and replaces it by a corner near the indicated position. A new hypothesis is generated and its height recomputed. If a side is indicated, the closest side of the existing hypothesis is found and moved to include the indicated position. Again, a new hypothesis and building model is constructed.

Correction of height is more complex. A wrong height is obtained by wrongly matching lines during height computation. In the view which the user interacts with, a building with the wrong height will still appear as having its roof in the correct place but the baseline will be wrong (in the other view, either the base or the roof or both may be wrong). As illustrated in Figure 13, the roof corner is correct in 2D, but incorrect in 3D due to its wrong z-coordinate inferred from automatic height calculation. The user may correct his error by clicking anywhere along the correct base of the building. The system first calculates the correct ground corner by intersecting the new base line (parallel to a roof line) with one of the projected walls. If we use this ground corner and the current hypothesis, the roof corner will project along one of wall lines but not at the correct roof corner position, as also shown in Figure 13. We now need to correct the height so that the projected and roof corners coincide. Note that this can not be done simply by scaling the projected line lengths as the projection does not necessarily preserve length ratios. For simplicity, we find the correct height by a binary search. During each iteration of the binary search, the system takes the middle height in current height range and uses it to calculate the 3D coordinate of the roof corner and projects it back to 2D. The error between the projected 2D value and the correct 2D value tells the system the direction in which the height range should be reduced. This iteration is repeated until certain accuracy is reached or the number of iteration exceeds the maximum number of iterations allowed. In the latter case, the system indicates that the clicked ground point is not acceptable.

These processes are illustrated by examples below. Figure 14 shows a detected building that is only partially

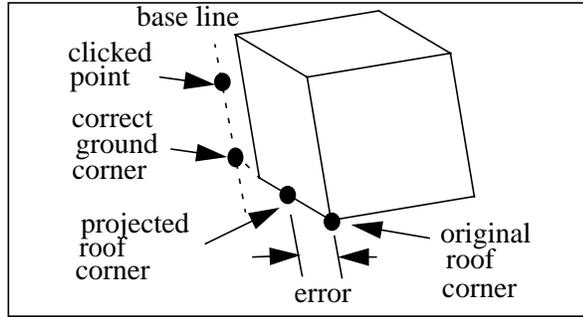


Figure 13 . Illustration of height correction.

correct. The appropriate correction consists of indicating a point along the actual building boundary to cause the system to adjust the incorrect side. As before, the system automatically recalculates the height and location of the new model.

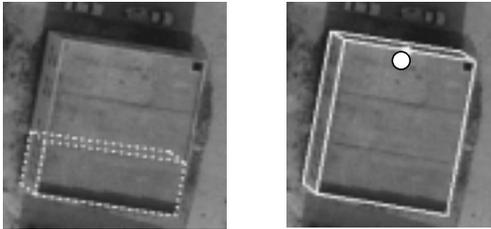


Figure 14 . Adjusting wrong side of a building

The last example, shown in Figure 15 illustrates a similar procedure to adjust the height of the building. The user needs only to select a point along the base of the building.

4 Results and Conclusions

This system has been applied to a number of images. Some results from portions of Ft. Hood, Texas site are shown in Figures 16 and 17 (the Ft. Hood site has been used by several researchers for automatic modeling in recent years and hence provides a good comparison point).

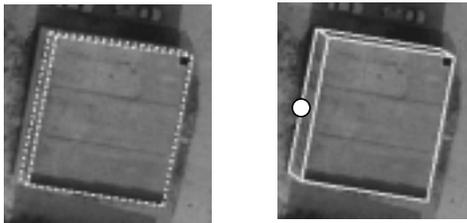


Figure 15 . Adjusting wrong height of a building.

Table 1 summarizes statistical information on the number of clicks and time needed to construct these models. For each roof type (flat or gable), the table shows the number of components that were detected using a given number of clicks. A total of 45 components were modeled, of which only 2 required subsequent height correction.

The total elapsed wall time for this example was 290 seconds. This time does *not* include the initial set up times,

such as for selecting an appropriate area to process) or for computing the features that the system uses for automatic analysis. Rather, only the actual time for modeling is included. We believe that these are meaningful parameters assuming that a large number of buildings are to be modeled. Our experiments with other images and other sites show similar time requirements.

Table 1: Distribution of Interactions for Three Areas from Ft. Hood Site

Roof Type	Clicks Needed	Components Formed
Flat Roof Buildings	1	1
	2	1
	3	14
Gabled Roof Buildings	2	5
	3	8
	4	16

The results show that the described system can be a very effective tool for constructing models for the classes of buildings within its capability. In our experience, it takes approximately one minute to model each building by using parameter fitting techniques such as provided in [7]. In [5], authors report that approximately 30 seconds are required to collect the points needed for recovering a roof description (though the roof types are more complex than for our system). We believe that our methodology can be extended to more complex shapes while still keeping the number of user interactions low; this will be a focus of our research in the future.

References

- [1] A. Gruen and R. Nevatia (Editors), Special Issue on Automatic Building Extraction from Aerial Images, *Computer Vision and Image Understanding*, November 1998.
- [2] A. Gruen, E. P. Baltsavias and O. Henricksson, *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*, Birkhauser Verlag, 1997.
- [3] S. Heuel and R. Nevatia, "Including Interaction in an Automated Modeling System", Proceedings of Symposium on Computer Vision, Miami Beach, FL, November 1995, pp. 383-388.
- [4] Y. Hsieh, "SiteCity: A Semi-Automated Site Modeling System", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, California, June 18-20 1996, pages 499-506
- [5] A. Gruen and H. Dan, "A Topology Builder for Automated Building Model Generation", in [2], pp. 149-160.



Figure 16 Results from Area 1 of the Fort Hood site.



Figure 17 . Results from Areas 2 and 3 in the Ft. Hood Site.

[6] S. Noronha and R. Nevatia, "Detection and Description of Buildings from Multiple Aerial Images", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, PR, June 1997, pp. 588-594

[7] T. Strat, L. Quam, J. Mundy, R. Welty, W. Bremner, M. Horwedel, D. Hackett, and A. Hoogs, "The RADIUS Common Development Environment", *Proceedings of the 1992 DARPA Image Understanding Workshop*, San Diego, California, pp. 215-222