Chapter 7
Assisted model construction

While the results produced by the automatic system described in the previous chapters are good, they are not perfect. In cases where the automatic system does not produce perfect results, such as a missed building (true negative), it still retains the feature hierarchy for the area covered by the true negative in each view. The domain knowledge possessed by the system, possibly in conjunction with the features detected in the area of the true negative, may be leveraged to produce a hypothesis corresponding to this building, with “hints” from a human user. When the system runs in the mode where it incorporates “hints”, or inputs, from the user, possibly using the precomputed feature hierarchy and domain knowledge, it is said to be in assisted mode.

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 as in the basic CME system [90]. 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 [33] [34]. As many urban areas contain buildings that are identical or very similar to others, tools for replicating them can also increase the user productivity as in [33].

Basic modeling tasks are still performed by the automatic system described earlier, but the system receives simple, yet critical, assistance from a user. The assisted system’s capabilities are limited by those of the underlying system. In this case, the shapes of the buildings are restricted to be rectilinear; the roofs may be either flat or symmetric gables. We describe two approaches that attempt to provide the human user with tools that augment the automatic system. The goals to be met are to significantly reduce user effort in the construction of models and to maintain or improve the quality of the results when compared
with the “dumb” manual systems i.e. systems that do not take advantage of geometric, photometric or domain-specific constraints.

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. Such interaction is called an “input”. The system requires two (or more) views of a scene with associated camera geometry; however, most user interactions take place in one view only. Other views may be displayed but the user is not asked to view the images stereoscopically. We believe that confining most of the interactions to one view can significantly reduce the effort required by the user.

Section 7.1 describes the “smart” real-time system which uses many of the geometric and domain-specific constraints available, and proceeds incrementally based on user inputs. Section 7.2 shows results obtained using the real-time user-assisted system and a performance comparison with a generic modeling tool. Section 7.3 concludes the chapter.

7.1 Real-time hypothesis generation based on user input

The approach detailed here constructs as building hypothesis in real-time based on user inputs. This approach uses the user’s input(s) as well as the context information, specifically lines and junctions, from the view, to form a plausible hypothesis. The presence of multiple views enables a 3-D hypothesis to be generated immediately. The process is geared to make it’s best guess based on the detected features and the available user inputs. This implies that as the user provides incremental input, the system should converge better on the desired output.

To determine a flat-roof rectangular hypothesis, a maximum of 3 positional inputs is required to determine the roof, with a possible additional input to correct the estimated 3-D height. This process is described in Section 7.1.1. Determining a symmetrical gable-roof hypothesis also requires a 3 positional inputs, with 2 possible additional inputs to correct the heights of the sides and the spine of the gable-roof. The method described in this thesis for gable-roof hypotheses requires 5 positional inputs in 2 views, which accurately determine the gable-roof with the correct 3-D heights. This process is outlined in Section 7.1.2.
7.1.1 Generation of flat-roof rectangular hypotheses from a single view

Flat-roof rectangular hypothesis formation requires a maximum of 3 positional user inputs in a single view, with a possible 4th input to correct the 3-D height of the generated 3-D hypothesis. Tests on a number of buildings reveal that fewer than 30% of the hypotheses needed 3 positional inputs, and none needed a height correction. The actions performed by the system after each input are outlined in the following paragraphs. Figure 7.1 is a flow-chart that diagrams the process. Figure 7.2 shows a view of building

![Flow-chart of hypothesis formation](image)

**Figure 7.1 Construction of flat-roof hypotheses in real-time**

that is used as an example to illustrate the user-assisted process.

**Actions after single input**

Figure 7.3 depicts the situation after the first positional input from the user. The system executes the following algorithm:
from the set of all junctions $S_{ji}$ for $i=1..n_{views}$, defined in Section 4.1, locate all hypothesized junctions near (within a radius of 5 pixels of) the positional input.

if no junctions are detected report failure and exit.

for each junction found, attempt to construct a parallelogram as follows: use lines forming the junction to derive the parallelogram (2-D roof hypothesis). Closures for the two other sides of the parallelogram are sought from the set of matched lines $S_{lm}$, defined in Section 4.1, using the procedure for finding closures for parallels in Section 4.2.1. The best closures (as defined in Section 4.2.1) determine the two other sides of the parallelogram. If no closure is found on one side, the side is hypothesized to begin where the junction leg adjacent to it ends, and is parallel to the side opposite it. If neither closure is found the junction is discarded and no hypothesis is generated for this junction.

match the 2-D parallelogram across the available views, and select the best match (as defined in Appendix A.) across all views, to yield a parallelogram match (building hypothesis).

compute verification scores for this building hypothesis using Equation (5.1), as is done during unassisted (automatic) operation.

select the best 3-D hypothesis (the one with the highest verification score) from the set of all 3-D hypotheses generated by junctions in the neighborhood of the user input and present it to the user.

Figure 7.2  Building used to illustrate user-assistance
if the user is not satisfied with the hypothesis, or if there is no hypothesis generated, allow the user to provide an additional positional cue to the system. Figure 7.4 illustrates the results on the example in Figure 7.2 after the first user input.

![Figure 7.3 Actions after the first positional input](image)

**Figure 7.3  Actions after the first positional input**

**Figure 7.4  Results after first input on building in Figure 7.2**

**Actions after second input**

Figure 7.5 illustrates the two cases that arise after the second input. The second input is used to generate new hypotheses in the same manner as with the first input. However, hypotheses whose projections have corners near the first input as well, are considered first. If no such hypotheses are found, hypotheses that are formed exclusively from the second input *i.e.* which do not have corners near the first input, are examined to retrieve the best hypothesis from that set. If the user is not satisfied with the hypothesis, or if no hypothesis is generated, the user may choose to provide a 3rd and final positional cue. Figure 7.6 shows the results after the second user input on the example in Figure 7.2.
Actions after third input

On receiving the third input, the following algorithm is executed:

- use the three points to form three possible parallelograms to represent roof hypotheses, as shown in Figure 7.7. The adjacent sides of the parallelogram must be projections of lines that are perpendicular in 3D and parallel to the ground. This constraint is applied after each of the parallelograms is matched in all the views.

- find the best parallelogram matches (building hypotheses) across all available views for each of the three hypotheses, using the process detailed in Appendix A..

- calculate the 3D orientation of the planes, for each of the three building hypotheses.
• select the hypothesis with least inclination to the ground plane.

The results after the third input on the example in Figure 7.2 are shown in Figure 7.8. After the third input, a result is always formed. The result is correct in the particular image where the 3 inputs have been specified. However, there 3D height of the results may not be estimated correctly. To handle a wrong 3D height estimation, the user may specify the height by pointing to one of the corners of the building on the ground. This additional height input specifies the building unambiguously. Figure 7.9 illustrates the points a user might specify as corners of the building on the ground. In tests carried out, this height correction was not required.

Figure 7.7 Three 2-D hypotheses possible from 3 positional inputs

Figure 7.8 Results after third input on building in Figure 7.2
7.1.2 Generation of gable-roof rectangular hypotheses

Gable-roof hypotheses are generated by providing enough input for the system to accurately determine a 3-D gable-hypotheses, using knowledge of the camera models, and without relying on the underlying evidence. The following process, which requires the user to input 3 positions in one image and 2 in another, completely specifies a gable-roof hypothesis. This required inputs are as follows:

- specify one endpoint, $p_{i1}$, of a side of the gable in one view, say $view_i$
- specify an endpoint of the spine, $p_{i2}$, nearer to the input in the previous step
- specify the other end of the spine, $p_{i5}$
- specify the point matching $p_{i1}$, say $p_{j1}$, in another view, say $view_j$
- specify the point matching $p_{i5}$, say $p_{j5}$, in $view_j$

Given these inputs the system executes the following algorithm to the gable-roof hypothesis:

- from $p_{i1}$ and $p_{j1}$, derive the 3-D point $P_1$ that projects to $p_{i1}$ in $view_i$ and $p_{j1}$ in $view_j$, using the camera models provided.
- similarly derive $P_5$, the 3-D point corresponding to $p_{i5}$ and $p_{j5}$ in their respective views.
- The spine of the gable-roof is defined by $P_2$ and $P_5$, and is assumed to be parallel to the ground in 3D. Thus $P_2$ is at the same world z-coordinate as $P_5$. A ray from the
camera center of view through $p_{12}$ is extended into the 3D world to intersect the plane parallel to the ground and at the same z-coordinate as $P_5$. This unambiguously defined $P_2$.

- given $P_1$, $P_2$, $P_5$, and the assumption that the gable-roof is symmetrical about the spine, in 3-D, reflect $P_1$ about line $(P_2, P_5)$ to determine $P_3$.

- the side of the gable-roof $(P_1, P_6)$ is parallel to $(P_2, P_5)$ and has the same length. This determines $P_6$.

- similarly compute $P_4$, using $P_3$.

This specifies the gable-hypothesis completely, and unambiguously, without the need for height correction. This entities described are diagrammed in Figure 7.10. Figure 7.11 depicts the process. Figure 7.12 shows a real example with the views with the user inputs and constructed model overlaid in each view.

The advantage of this method over the method used for flat-roof hypotheses is that it does not need any feature information, or geometric constraints of the model. The reader is referred to [53] for a method for gable-roof hypotheses that takes advantage of the detected features, often reducing the need for precise input from the user.

### 7.2 Real-time Assisted Results

The real-time system for assisting the user relies on many of the techniques developed for the automatic system. As a result, it is able to generate good results with relatively a relatively small number of simple inputs. Figure 7.13 shows the model constructed using the real-time assisted system. As this system allows to user to override its hypotheses at each stage, the user is able to construct a model which can be as accurate as one constructed by a system that does not possess the mechanisms for utilizing detected features in the hypothesis formation process. Hence accuracy of the model is not an issue in the evaluation of this system. One measure of the utility of the real-time assisted system is the time it could save over constructing the model using standard model-construction tools such as those provided within the Radius Common Development Environment (RCDE). The other measure is the number of inputs required in each system to achieve the same result. These comparisons are provided in the following paragraphs and summarized.
Comparison of time taken: Using the real-time assisted system, the model for Fort Benning (in Figure 7.13) was constructed in 375 seconds, or 6 minutes 15 seconds. The same model, when constructed in RCDE by a user familiar with the constructions tools, took in excess of 20 minutes. Some interesting differences were the
computation of the height of the “spine” of the gable-roof buildings and the accuracy of the inputs required. Using the real-time assisted system, providing the three inputs in one view and two more in another view automatically determined the “spine” of the gable-roof building in 3-D. No height adjustment was necessary. Using the RCDE modeling tools, each vertex was automatically created at the same 3D height. This necessitated adjustment of the height of the two ends of the “spine” of the gable as an extra step. This step required delicate manipulation, and was the most time-consuming. In the detection of flat-roof buildings, the real-time assisted system allowed greater error in user input as it searched for features in a neighbor-

---

**Figure 7.11**  *Generation of gable-roof hypotheses in assisted mode*
hood of the input. Further, in the cases that the system did require three user inputs to determine a flat-roof hypothesis, it utilized the camera geometry and the assumption that the buildings were rectilinear to constrain the shape of the projection of the hypothesis, and find the best fit for the three inputs.

- **Comparison of number of inputs:** The real-time assisted system required an average of 2 inputs for the 7 flat-roof buildings and 5 inputs for each of the 19 gable-roof buildings for a total of 109 inputs. Using the RCDE system, flat-roof buildings needed 4 inputs for the corners and 1 input to adjust the height. Gable-roof buildings
Figure 7.13  Real-time assisted results on Fort Benning
required 6 inputs to identify the corners and 2 inputs to adjust the height of the spine. The total using this method was 187 inputs.

7.3 Conclusion

The assisted mode of operation allows the user to rapidly build, or correct the model generated by automatic system. The goals of reducing the effort expended in constructing a model, even when done entirely in the assisted mode, is easily met, when it is compared with the time taken to construct a model without these tools. A detailed comparison of this process with the “dumb” (dependent entirely on user input, and which does not utilize geometrical, photometric and domain-specific constraints) was provided in Section 7.2.