Large-Scale 3D Reconstruction: A Triangulation-Based Approach

by

George Tao-Shun Chou

B.A. in Biophysics, University of Pennsylvania, 1991
B.S.E. in Computer Science, University of Pennsylvania, 1991
S.M. in Electrical Engineering and Computer Science, M.I.T., 1993

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology

August 2000

Author ______________________________
Department of Electrical Engineering and Computer Science
August 1, 2000

Certified by ______________________________
Seth Teller
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by ______________________________
Tomas Lozano-Perez
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by ______________________________
Arthur C. Smith
Chair, Departmental Committee on Graduate Students

Large-Scale 3D Reconstruction: A Triangulation-Based Approach

by

George Tao-Shun Chou

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Abstract

Large-scale 3D reconstruction is a challenging engineering problem. For any system designed to address this problem, it is important to have these three capabilities: correspondence in 3D, incremental processing, and surface generation. In this thesis, we present an incremental, hypothesis-based approach to solving the large-scale 3D reconstruction problem. Images with accurate camera pose information are used as input. Our method begins with a systematic generation of correspondence hypotheses. For each hypothesis, we estimate its hypothetical structure through a multi-image triangulation process. Using the results of triangulation, a posterior probability for the validity of each hypothesis is computed. The computation is based on how well the supporting features match up in 3D space, and how many supporting features each hypothesis has.

The posterior probability is then used as a fitness measure for selecting the most probable correspondence hypotheses. Hypotheses compete to gain the support of features. When the probability of a hypothesis exceeds a certain level, the hypothesis is confirmed and becomes a permanent 3D structural element. When the probability drops below a predetermined threshold, the hypothesis is eliminated. Based on confirmed vertices and line segments, surface hypotheses are constructed. Confirmed surfaces provide visibility constraints that are then used to eliminate unlikely feature correspondence hypotheses. We assume that all surfaces of interest are planar and opaque. Results of reconstructing buildings in the Technology Square from real and synthetic images are presented. The observed run-time of the system is approximately a quadratic function of the number of input features.

Thesis Committee:	Professor Seth Teller
	Professor Tomas Lozano-Perez
	Professor William M. Wells III

Acknowledgements

First of all, I would like to thank my advisors and committee members for their supervision, and for giving me the opportunity to pursue the research described in this thesis. Without their guidance and confidence in me, this thesis would not have been possible.

Next, I would like to acknowledge my friends and colleagues in Artificial Intelligence Laboratory, in Laboratory for Computer Science, and at Massachusetts Institute of Technology, for their friendship and support over the years.

I also would like to express appreciation of my past teachers – in universities, in secondary schools, in grade schools, in the United States, and in Taiwan. They have all played an unforgettable role in my education.

Finally, I would like to pay a special tribute to my wife for her patience and sacrifice. It has been quite a journey for both of us. I also want to thank my parents and siblings who have supported us along the way.

Funding for this research was partly provided by the Advanced Research Projects Agency of the U.S. Department of Defense.

Contents

Chapter 1

Introduction

It has long been recognized that automatic 3D reconstruction is an important enabling technology for realistic, large-scale simulation/visualization of the world. Realism in simulation is enhanced by accurate modeling of the environment being simulated. To obtain the requisite 3D models, a variety of computerized methods have been developed, ranging from active methods such as laser scanning, structured lighting, to passive methods such as structure from stereo, structure from motion, structure from texture/shading, structure from contour/line-drawing, to combinations of techniques.

Despite tremendous research efforts, automatic acquisition of 3D models from 2D imagery remains a challenging problem. Most existing methods that operate on real images still require a significant amount of human intervention. For a large-scale project, the labor involved can be quite prohibitive. This impediment provides the main motivation for the present thesis – to develop a fully automatic 3D reconstruction system using visual images.

A typical reconstruction scenario encountered in our research is illustrated in Figure 1.1. The input to the system consists of several digitized images, taken with different camera positions and orientations. In this thesis, we shall refer to position, orientation and other essential camera parameters collectively as the camera pose. Two of the three input images are shown in Figures 1.1 (a) and (b). Significant line segment and vertex features in the input images are extracted by an automatic process, and the results are displayed in Figures 1.1 (c) and (d). Then through a triangulation process that computes the intersection of features back-projected into 3D space, parts of the desired 3D model are formed. In this thesis, we shall refer to feature back-projections as *extrusions*.

To recover 3D structure in the scene from 2D images, there are three general problems:

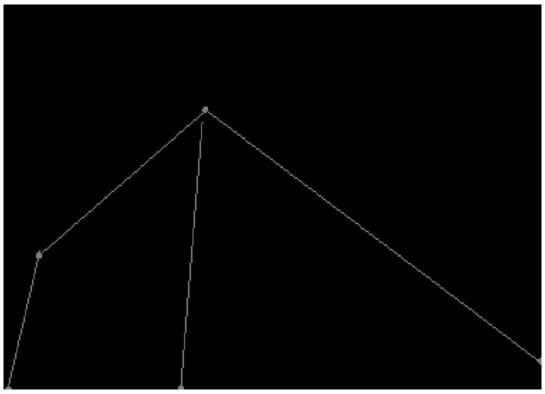
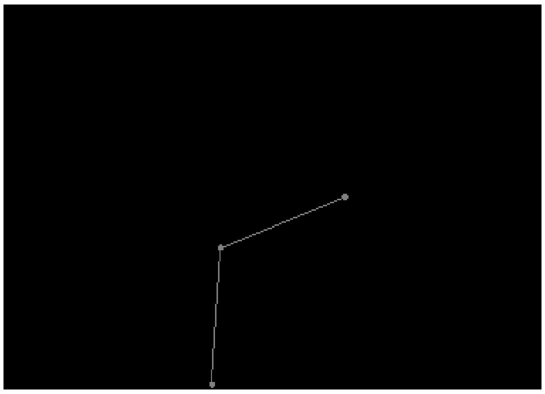
1. Determination of camera pose – obtaining camera motion, relative camera pose, or absolute camera pose for each image.

2. Computation of correspondence – matching image intensity, low-level features, or high-level structures, from different images.

3. Construction of 3D descriptions – computing dense depth maps, connected line segments, or surface-based/object-based models.

Depending on the problem formulation, these tasks may not be independent of one another. In fact, solving one problem can simplify the task of another. For example, determining reliable feature correspondences is often helpful for obtaining accurate camera poses, and vice versa. Likewise, recovered high-level scene descriptions may be used to improve the reliability of low-level feature correspondences, and vice versa. Managing the interdependence of these sub-problems is a major challenge in 3D reconstruction systems.

(a)  (b) 

(c)  (d) 

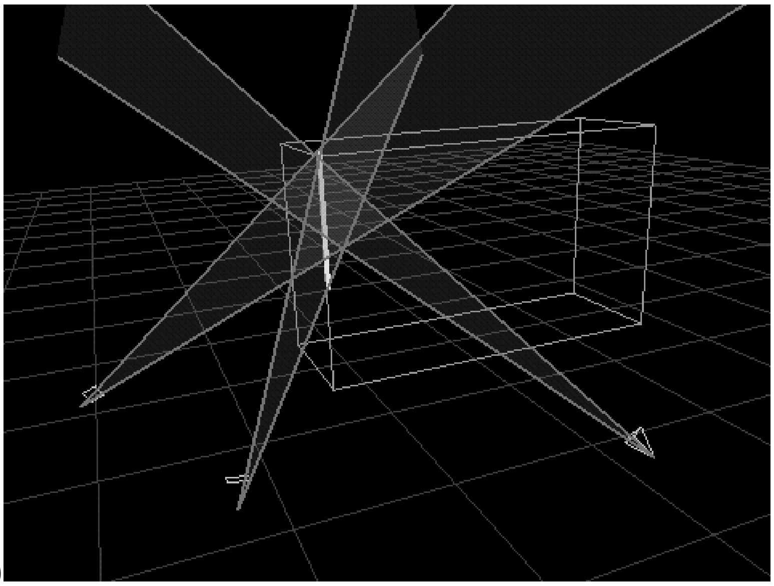
(e) 

Figure 1.1: A typical reconstruction scenario encountered in our research. (a)(b) Grayscale version of two of the color input images. (c)(d) Significant line segment and vertices extracted from the two input images. (e) The 3D triangulation process.

1.1 Previous Work

Existing work in 3D reconstruction from 2D images can be characterized by their selective attention to the three basic problems, and how they attempt to address them.

The earliest studies focused on solving the correspondence problem in a stereo matching framework. Some of the original motivation came from research into human stereo vision [35]. A successful strategy developed by researchers is the *coarse-to-fine* method [24], [34], [47]. In these systems, matching begins at a low resolution of the image in order to cover large displacements. Matching then proceeds to higher resolutions where results from lower resolutions are used to constrain the search. The main drawback is that these methods cannot deal with significant viewpoint changes.

The next step is to extend correspondence operation to multiple images. The simplest method is to use images acquired from closely spaced cameras along a predetermined path [26], [6], [5], [37], [45]. Given the temporal coherence of these images, the correspondence problem can be viewed as a tracking problem in pixel intensity or image features. Central to most of the proposed systems is the application of Kalman filtering. This tracking algorithm has proven to be very effective in dealing with random noise, but less so with image occlusion.

If the images are taken by a freely moving camera, one must estimate the camera motion, or relative pose of images, in addition to correspondence. This is known as the *structure-from-motion* problem [32], [1], [21], [49], [25], [16]. A typical algorithm begins by estimating the motion flow or image/feature correspondence, solves for the camera motion or relative pose, and then computes the 3D structure. The results of these systems with real images appear to be qualitatively correct. Analogous to methods in the previous category, these methods also require densely sampled images as input to the system, for ease of motion estimation and tracking.

Due to difficulties in analyzing images of man-made environment, which often contain sparse features, researchers began to match more complex scene descriptions [3], [31], [7], [28], [11], [48]. It has been observed that higher structural properties tend to be more stable with respect to viewpoint changes than low-level image properties. Thus the structural relationship between image features can be used for resolving matches. However, the key process of linking up relevant features into self-consistent models is usually ad-hoc and unreliable.

Recently, several algorithms capable of analyzing long baseline inputs, or images from distant viewpoints have been proposed [43], [8], [12], [18], [14], [38]. In these systems, the camera pose information is assumed to be given. Nevertheless, the correspondence problem is made more challenging due to the potential for significant viewpoint changes. One feasible solution involves performing a space-sweep search in the 3D space for clusters of observed evidence [12], [14], [38]. Interesting results from real images have been demonstrated with this approach.

1.2 Large-Scale Reconstruction

Our objective is to develop a large-scale reconstruction system capable of producing 3D models suitable for use in realistic visualizations. By large-scale, we mean a system that

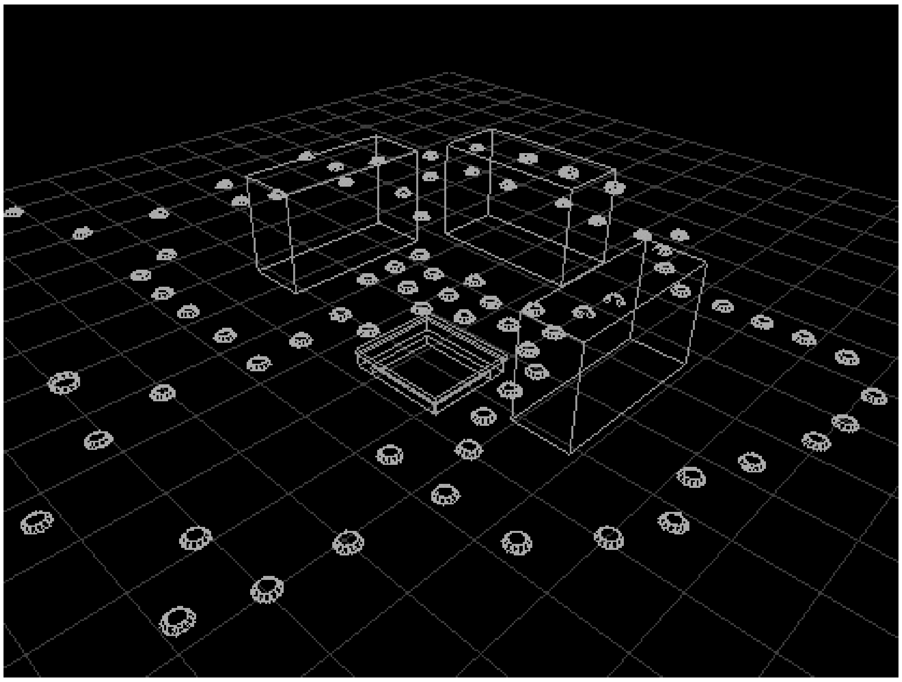


Figure 1.2: Virtual view of a large-scale reconstruction project around the Technology Square at MIT. Each node on the ground represents a cluster of 46 images acquired from that position. The wireframes represent models of buildings to be analyzed.

can easily handle hundreds or more input images, acquired from multiple instances in time and space [46]. See Figure 1.2. Large-scale reconstruction systems have different requirements from the traditional smaller-scale systems. We consider the following capabilities important in the design of such a system:

1. Correspondence in 3D – key to obtaining correspondence between images with significant viewpoint changes.

2. Incremental Processing – key to system scalability, system transparency, and ease of debugging.

3. Surface Generation – a more useful output format, and facilitates the low-level correspondence process.

In large reconstruction projects, the need for processing and storing large amounts of image data can be tremendous. For reconstruction systems facing such tasks, the ability to form correspondence on images taken from distant viewpoints is a powerful one because it can mean collecting and processing fewer images. Most existing systems do not have such capability. As a result, large amounts of image data are required even for a small-scale project, and the efficiency of output generation is low. However, if the correspondence problem can be solved directly in the 3D space, then the input images can be more disparate, and fewer images are needed.

For large-scale reconstruction systems, the ability to easily update and reveal the current state of reconstruction is important for two reasons: scalability, and ease of debugging. In general, a reconstruction system can either operate in an incremental mode or a batch mode. A batch system outputs a final model at the end of processing all the input images. On the other hand, an incremental system takes in one image at a time, and modifies the model with each input image. By not having to re-process all the old images with each additional image, an incremental system has the advantage of being more scalable, and transparent with respect to the input. Therefore, we would like our system to process input in an incremental fashion.

For any reconstruction task, the ability to generate mid-level representation in the form of 3D surfaces is desirable. Many existing systems can only generate low-level descriptions such as floating 3D vertices and line segments. There are two good reasons for constructing surfaces [10]. First, surface representations are much more useful than low-level line/vertex-type representations in visualization and most other kinds of subsequent processing. Second, information from recovered surfaces can provide important visibility constraints for the correspondence process by reducing matching ambiguity. Motivated by these advantages, we have developed a system which simultaneously processes information at multiple levels of representation.

Another consideration, not listed above, but nonetheless is important for us, is the ability to process input images in arbitrary order, without placing assumptions on the order of acquisition or constraints on the viewpoint [9]. Such flexibility can enable us to perform 3D reconstruction in a spatially focused manner, where only images viewing a designated 3D region in space are selected for processing. The benefit is that we will be able to update the reconstruction result for any 3D region in space very efficiently. Thus, instead of imposing order on the input, ordering will be imposed on the output.

1.3 Correspondence via Triangulation

The method of triangulation is a well-known method in the fields of astronomy and machine vision. It is based on the simple, yet powerful idea that using measurements taken from two different locations A and B, the relative position of a distant object X can be estimated (see Figure 1.3). Specifically, given the baseline distance d between observation points A and B, and the corresponding viewing angles α and β, trigonometry tells us that the distance from baseline to the object X is $z = \dfrac{d\ \sin\alpha\ \sin\beta}{\sin(\alpha+\beta)}$.

In 3D reconstruction, the standard approach is to apply the triagulation method to estimate the 3D structure of a scene, *after* feature correspondence has been estab-

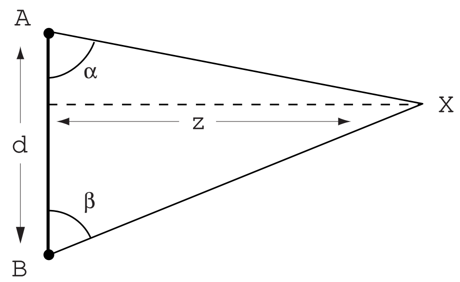


Figure 1.3: The original 2D triangulation problem.

lished. The goal of the correspondence operation is to ensure that only measurements corresponding to the same 3D object are used in estimating the structure. Implicit in this approach is the assumption that the correspondence problem can be solved in some way without knowing much about the scene structure. Feature or intensity matching in 2D image space is the typical solution.

Matching 2D features or intensity is attractive for its limited search space and relative ease of implementation. In particular, if we know the exact pose of the camera, the search can be restricted to constraint lines known as the *epipolar lines* [24], [33], [22]. However, 2D matching requires that the appearance and the positions of the features do not change significantly between matching images. This is true only when the viewpoints of matching images are not very far apart. As a result, systems based on 2D matching cannot work on non-adjacent or distant images.

To handle significant viewpoint changes in the input, a reconstruction system should use feature properties that are invariant with respect to camera motion to form correspondence. Only properties intrinsic to the imaged objects, such as 3D geometry or

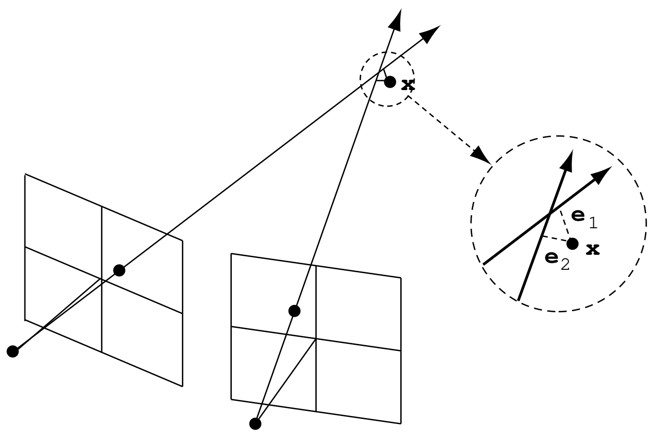


Figure 1.4: The general 3D triangulation problem. Due to noise and measurement error, the intersection process will generate error residuals.

structure, meet this criterion. This means that we are faced with the difficult challenge of solving for the correspondence problem and 3D structure *simultaneously*. In addition, due to noise and measurement error, the general 3D triangulation approach will not produce unique intersections (see Figure 1.4).

To see how this dual problem may be addressed, one needs to recognize that the 3D triangulation process can provide not just estimates for the structure of the corresponding features, but also information on the quality of correspondence. The error measure of the estimation process gives us an indication on how well the features match up in 3D space. If the residual is large, one can infer that there is no single 3D structure near which all of the features intersect. In such a case, the presumed correspondence should be questioned. If the residual is small, our belief in the 3D estimate and the correspondence is stronger.

From this perspective, the triangulation method should be viewed as an important tool suitable for solving the dual problem of structure estimation and feature correspondence. In our reconstruction system, the triangulation method plays this role.

1.4 Overview of the System

In this thesis, we present an incremental, hypothesis-based approach to solving the reconstruction problem. Our method begins with a systematic generation of correspondence hypotheses. For each hypothesis, we estimate its hypothesized structure through a multi-image triangulation process. Using the results of triangulation, a posterior probability for the validity of each correspondence hypothesis is computed. The computation is based on how well the supporting features match up in 3D space, and how many supporting features each hypothesis has.

The posterior probability is then used as a fitness measure for selecting the most probable correspondence hypotheses. When the probability of a hypothesis exceeds a certain level, the hypothesis is confirmed and becomes a permanent 3D structural element. Un-confirmed hypotheses stay in the system as long as their posterior probabilities remain above a certain level. But when the probability drops below a pre-determined threshold, the hypothesis is eliminated.

New images are not required to be inserted in any particular order. With each image, new hypotheses are formed and existing hypotheses are encouraged to grow by forming links with new features consistent with their hypothesized 3D structures. The result is a web of links between image features and 3D hypotheses (see Figure 1.5). We keep track of the 3D estimate and posterior probability for each correspondence hypoth-

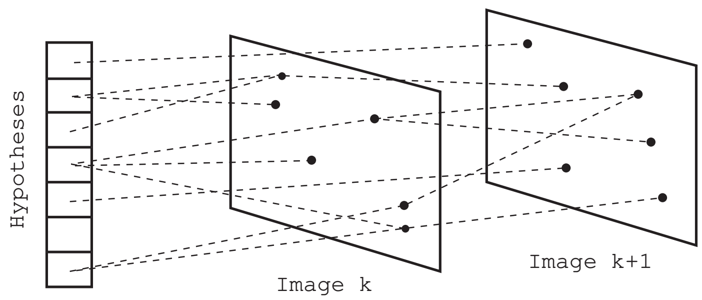


Figure 1.5: Links between image features and correspondence hypotheses

esis and update them whenever necessary. Through repeated hypothesis generation, triangulation, and selection, we gradually assemble a set of most probable correspondence hypotheses.

While correspondence hypotheses are being processed and 3D elements are being formed for low-level vertices and line segments, similar processing is happening at the mid-level for surface representation. Surface level hypotheses are built upon confirmed elements of 3D vertices and line segments. All levels of representations are computed as concurrently as possible, so that the latest surface information can be used to correct and refine correspondences made at the vertex and segment levels.

We focus our system on analyzing urban environments where there is an abundance of vertex and line segment features. For each image, the input to our system consists of information on the position, orientation, focal length, and other parameters of the camera, along with two types of image features: 2D segments – extracted by fitting lines to the output of an edge detector, and 2D vertices – located by intersecting 2D segments that form L-junctions.

Finally, we assume in this thesis that accurate and complete camera pose information is available to us. Advances in global positioning system (GPS) technology is rapidly simplifying the task of obtaining accurate camera pose information. Currently, the pose information obtained through existing GPS technology still needs to be refined by a process involving feature correspondences before it can be put to use [13], [2]. However, improvement in GPS accuracy should make pose refinement simpler in the future. This is a significant technological advantage which we have over earlier systems.

1.5 Thesis Organization

The remainder of this thesis is organized as follows.

In Chapter 2, we will provide an overview of our correspondence system. We first examine the correspondence problem in the context of a triangulation-based approach. Using two examples, we demonstrate the necessity of treating all initial matches as hypotheses that must be confirmed by additional evidence. We then proceed to define the main data structures used in our system, and outline the correspondence algorithm with a flow diagram.

In Chapter 3, we will present our triangulation methods for correspondence. We begin by describing the hypothesis formation process that leads to triangulation. Then we derive two multi-image triangulation methods for computing the intersections of vertex extrusions and segment extrusions in 3D space. The methods are developed in a probabilistic framework to highlight the uncertainties involved. Special situations under which these methods may fail are also noted.

In Chapter 4, we will examine the processing of correspondence hypotheses. The computation of posterior hypothesis probability involves two components: the data consistency probability, which is a function of triangulation residual, and the non-accidentalness probability, which is a function of the number of supporting features. Based on posterior probability, a hypothesis may either be confirmed, updated, or rejected. We outline the chain of events that occurs in each case.

In Chapter 5, we will describe the computation of surfaces in our system. The two major tasks involved in surface computation are: surface hypothesis processing, and enforcement of the visibility constraints of confirmed surfaces. These tasks are accomplished under the assumptions that the surfaces of interest can be effectively modeled by piece-wise planar and opaque polygons. Maximum likelihood estimators for the parameters of planar surfaces are derived.

In Chapter 6, we will illustrate the function and performance of our system. The graphical user interface of the system is introduced to demonstrate how one interacts with the system. We then present reconstruction results of our system operating on two large data sets, one consisted of synthetic images, and the other real photographs. Our analysis shows that the visibility constraints of confirmed surfaces play an important role in improving the correspondence process.

In Chapter 7, we will conclude the thesis by taking another look at our system and the problem of 3D reconstruction. First, the main contributions of our system, as well as some of the challenging areas are itemized. Then to inspire new ways of looking at the problem, theoretical connections with other fields are contemplated. Finally, we summarize the entire thesis.

Chapter 2

Correspondence System

Due to the efficiency and scalability requirements of handling large-scale 3D reconstruction projects, our system must be able to address the dual problem of correspondence and reconstruction in an incremental manner. We propose a triangulation-based approach for finding the most probable correspondence hypotheses based on the following three-step process:

1. Generate correspondence hypothesis

2. Perform triangulation on the hypothesis

3. Evaluate correspondence hypothesis

Our strategy is to generate all plausible correspondence hypotheses, and then test each one through 3D triangulation. In this chapter, we will first examine the correspondence problem in the context of a triangulation-based approach. We will then advance our method by defining the main data structures, and describing how the prescribed three-step process is embedded in our system. Specific details for the three steps listed above will be presented in the next two chapters.

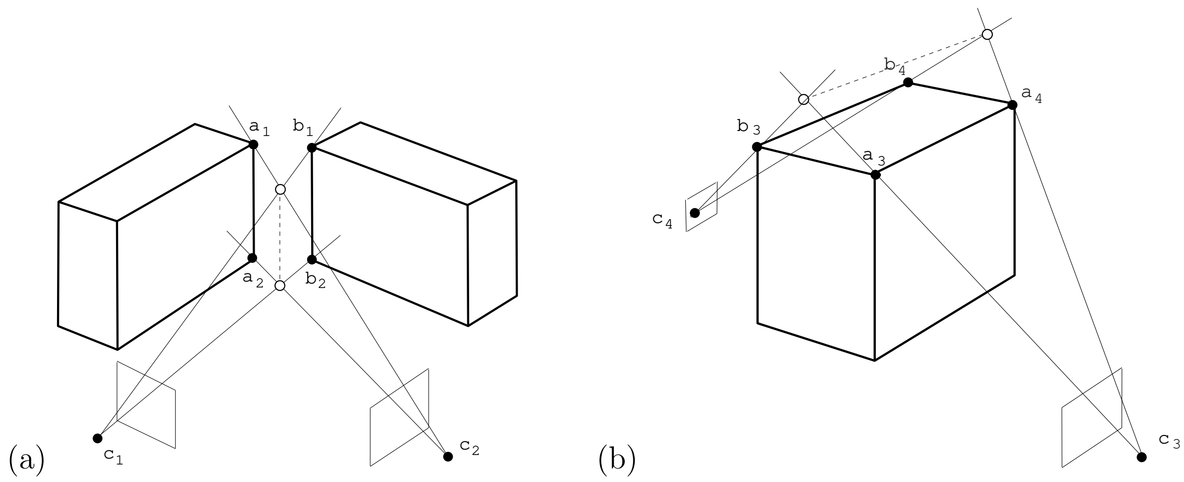


Figure 2.1: The correspondence problem in a triangulation-based approach: two examples of accidental intersections of extrusions.

2.1 The Correspondence Problem

In traditional stereo or 3D reconstruction research, the *correspondence problem* refers to the ambiguity in making feature correspondence due to multiple potential matches [36]. In a triangulation-based approach, we face the same problem, but in a slightly different form. Let us look at some of the potential difficulties, and consider how the correspondence problem may be addressed.

In a triangulation-based approach, the correspondence problem often shows up in two forms. In the first form, the observed feature is interpreted as closer than its true position. In the second form, the observed feature is interpreted as further than its true position. Figure 2.1 demonstrates both forms.

In the left part of Figure 2.1, the extrusions of vertices a_1 and a_2 intersect with the extrusions of vertices b_1 and b_2 in front of the two buildings. In this case, the intersections occur *in front* of the true position of the 3D structures. The resulting structure could be interpreted as a line segment floating in front of the two buildings. In the right part of Figure 2.1, the extrusions of vertices a_3 and a_4 intersect with the extrusions of vertices b_3 and b_4 over the roof of the building. In this case, the intersections occur *behind* their true 3D positions, which could result in an incorrect interpretation for the shape of the roof.

Both of these cases are clearly incorrect, and yet they are not uncommon. We have noticed that whenever multiple images are taken with a camera revolving around some region in space, there will be many near-crossing extrusions that result in accidental intersections [9]. Moreover, we can never be certain that an extrusion associated with a feature is not an artifact of the feature extraction process. As a result, we should not hastily accept any apparent close intersection of extrusions as a good match, even when the triangulation error is small.

To resolve the ambiguity, additional evidence is needed. We observe that it is very unlikely for extrusions from three or more viewpoints to intersect at exactly the same position in 3D space. Hence, with more features intersecting at the same position, the probability that a 3D structure does exist there increases significantly. This provides a probabilistic basis for the confirmation of a 3D structure. However, we note that even when the aggregate probability for the existence of the 3D structure is strong, it does not mean that individual probabilities for each feature belonging to the same 3D structure is equally as strong. We may have one or more mis-matches among the good matches. Thus, we have to examine the correctness of every individual match.

We also note that certain structural properties, such as adjacency, are relatively invariant with respect to small changes in the viewing direction. In order for two vertices to match, one may require that two or more of their incident edges should match up also. Unfortunately, matching features jointly will be much more difficult than matching features individually, given the unreliable nature of existing feature detection systems. Features that are obvious to human eyes are frequently missed by feature detection systems. Moreover, incorrect feature artifacts may get generated by the feature detection systems, unexpectedly.

2.2 System Data Structures

We now turn our attention to the proposed correspondence system. In our system, there are three basic types of data structures:

- **Vertex type** – any data structure related directly to the construction and representation of a 3D point. In the input image, each vertex feature is detected as the L-junction of two connected line segments.

- **Segment type** – any data structure related directly to the construction and representation of a 3D line segment. In the input image, each line segment is extracted by fitting a straight line to the output of edge detection.

- **Surface type** – any data structure related directly to the construction and representation of a 3D planar surface. It is constructed from confirmed vertex and segment elements.

In addition to the previous four classes, data structures in our correspondence system are further categorized into the following four classes:

- **Features** – these form the main input to the correspondence system. There are two types of features – vertices and line segments. Each vertex feature is represented by its image position, its incident lines, and links to its competiting vertex hypotheses. Each line segment is described primarily by the two end points, and links to its segment hypotheses.
- **Extrusions** – are back-projections of features through their corresponding camera centers into 3D space. For a vertex extrusion, it is represented as a 3D ray anchored on a camera center. For a segment extrusion, it is represented as a 3D wedge spanned by two rays bounding the line segment, anchored on a camera center. See Figure 2.2 for a visualization of vertex and segment extrusions.
- **Hypotheses** – are hypothetical 3D structures computed through triangulation of feature extrusions of the same type, except for the surface hypotheses, which are derived from vertex and segment elements. In general, a 3D hypothesis is represented by a 3D estimate of its structure, its intersection errors, and links to its supporting features.
- **Elements** – are confirmed hypotheses. Each element is described in the same way as a corresponding hypothesis, except that once a hypothesis becomes an element, it will persist in the system.

We note that vertex and segment types appear in all four classes of data structures (feature, extrusion, hypothesis, element), while the surface type appears only in two classes (hypothesis, element).

In an ideal case, every feature should be assigned to exactly one 3D structure, while each 3D structure will be supported by multiple observations. This corresponds to a many-to-one mapping between the features and hypotheses, as illustrated in Figure

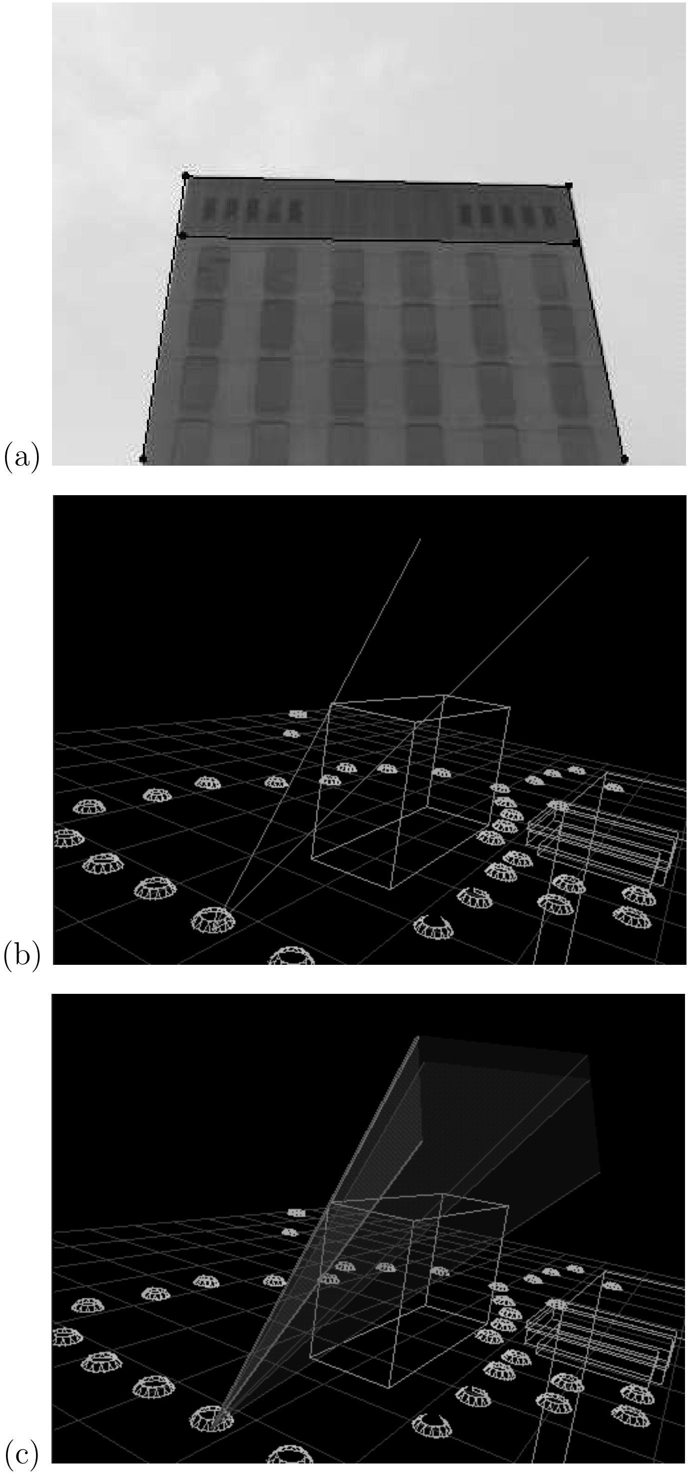


Figure 2.2: Image features and extrusions: (a) Significant features extracted from an image. (b) Visualization of the corresponding vertex extrusions. (c) Visualization of the corresponding segment extrusions.

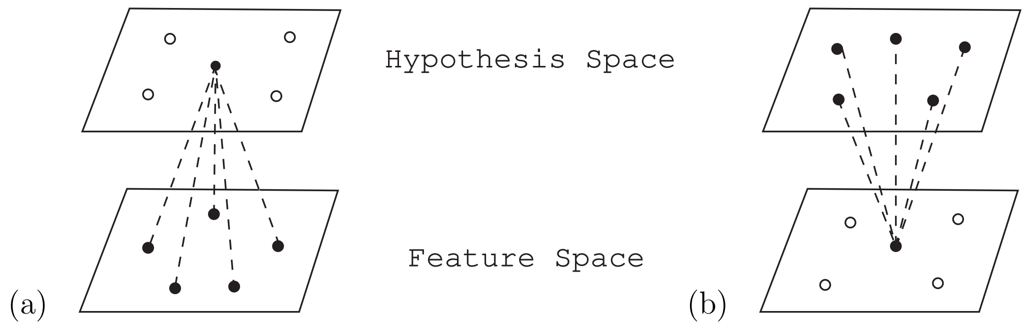


Figure 2.3: Feature-hypothesis connections: (a) Supporting features of a hypothesis, (b) Competing hypotheses of a feature.

2.3(a). The set of observed features associated with a hypothesis is referred to as the **supporting features** of the hypothesis.

But in reality, due to the uncertainty of feature correspondence, the mapping between features and hypotheses is usually not many-to-one but many-to-many. This means each image feature may be linked with a number of existing 3D hypotheses, as illustrated in Figure 2.3(b). The set of potential hypotheses associated with a feature is referred to as the **competing hypotheses** of the feature.

Once a hypothesis has been promoted into an element, all of its supporting features shall link only with the confirmed element. Links with all other hypotheses are severed. The supporting features of an element is thus referred to as the **committed features** of the element. The objective of our correspondence system is to convert the feature-hypothesis mapping from many-to-many to many-to-one by promoting hypotheses into elements. In the process, many competing hypotheses will be eliminated.

2.3 Incremental Processing

The proposed system processes input features and images incrementally without any ordering constraint. As images are inserted into the system, the algorithm searches for new features that are consistent with existing hypotheses. Concurrently, new hypotheses are generated from consistent features. The incremental 3D reconstruction problem can hence be viewed as a tracking problem, where 3D hypotheses are generated and tracked over images that *may* or *may not* be temporally coherent.

Incremental correspondence algorithms can be categorized by their tracking update methods. Bar-Shalom & Fortmann have identified three general classes of algorithms: the Nearest-Neighbor Standard Filter, the Probabilistic Data Association Filter, and the Track Splitting Filter [4]. Their classification is interesting, but not comprehensive. We propose the following three classes as more comprehensive, and reflective of the relationships between the classes:

1. Linear Exact Match (LEM) Algorithm – is the simplest incremental correspondence algorithm. It determines one of the features in the new image as the optimal match, and uses its information for update. The nearest neighbor algorithm is a member of this class. The algorithm rejects all other features as noise.

2. Linear Mixed Match (LMM) Algorithm – is a more complex incremental correspondence algorithm. Every feature which comes within a certain range is considered as a potential match, and the features are tracked *collectively*. A mixture of information from these features is then used for update. The weighted average algorithm is a member of this class.

3. Branching Exact Match (BEM) Algorithm – is the most involved incremental correspondence algorithm. Every feature which comes within a certain range is

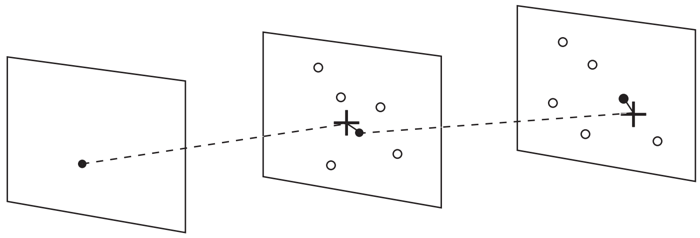


Figure 2.4: Linear Exact Match Algorithm

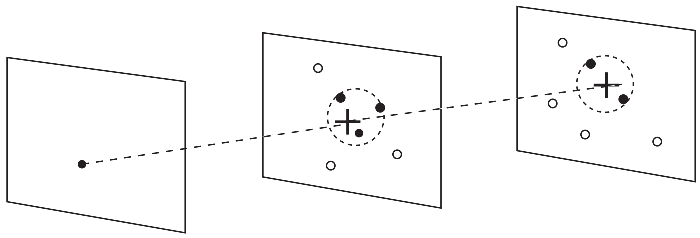


Figure 2.5: Linear Mixed Match Algorithm

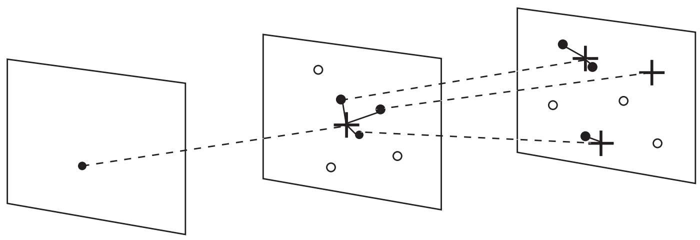


Figure 2.6: Branching Exact Match Algorithm

considered as a potential match, and the features are tracked *individually*. The hypothesis splits with every potential match feature. The split track algorithm is a a member of this class.

The algorithms are first distinguished by whether they track a single feature per image, or a mixture of features, for each active hypothesis. The LEM class locks onto a single feature in each new image, forming an exact match while ignoring all other features. The LMM class would use information from a subset of image features and form a match with a mixed combination of these features. The BEM class is also considered as an exact match algorithm, because for each tracking path, an exact match is made for each image.

The second attribute with which the algorithms are distinguished is by whether they could split the hypothesis tracking path or not. When encountering a new image with multiple potential matches, the first two classes of algorithms do not generate new hypotheses, they merely update existing hypotheses. They are considered as **linear path** methods. On the other hand, the third class of algorithm has the capability of pursuing multiple potential matches for each 3D hypotheses under tracking. It is therefore referred to as the split track or **branching path** class. See Figure 2.6.

In the current system, only the linear exact match algorithm has been fully implemented and tested.

2.4 Matching Algorithm Outline

Upon initialization, with no prior knowledge about the correct feature correspondence, our only strategy is to generate all plausible correspondence hypotheses, and then test them one by one. After the system has been running for a while, we are no longer in the dark. In fact, we can expect that some, if not many of the new features will match up with existing correspondence hypotheses or elements.

As features are inserted into the system, new hypotheses are generated at an extremely high rate. To control the computational complexity, the number of hypotheses generated need to be minimized. One way of achieving this effect is to order the sequence of hypothesis formation and testing processes following the insertion of each new feature into the system. The most efficient processing can be achieved by testing correspondence hypotheses composed of the new feature and existing elements first, following by hypotheses composed of the new feature and existing hypotheses, and lastly, hypotheses composed of the new feature and other extrusions.

This ordering is illustrated in Figure 2.7, a flow diagram of the feature correspondence process. In the diagram, the abbreviation "ST" stands for short-term or temporary hypothesis, while "LT" stands for long-term or tracking hypothesis. Each of the following labeled steps correspond to an identically labeled box in the flow diagram:

A. Construct a new vertex or segment extrusion.

B. Check to see if there is any element of the same type in the system that has not been tested with the new extrusion.

C. Check to see if there is any LT or long-term hypothesis of the same type in the system that has not been tested with the new extrusion.

D. Check to see if there is any extrusion of the same type in the system that has not been tested with the new extrusion.

E. Check to see if there is another new extrusion in the input.

F. Form a temporary ST or short-term hypothesis by combining an existing extrusion with the new extrusion.

G. Form a temporary ST or short-term hypothesis by combining an existing LT hypothesis with the new extrusion.

H. Form a temporary ST or short-term hypothesis by combining an existing element with the new extrusion.

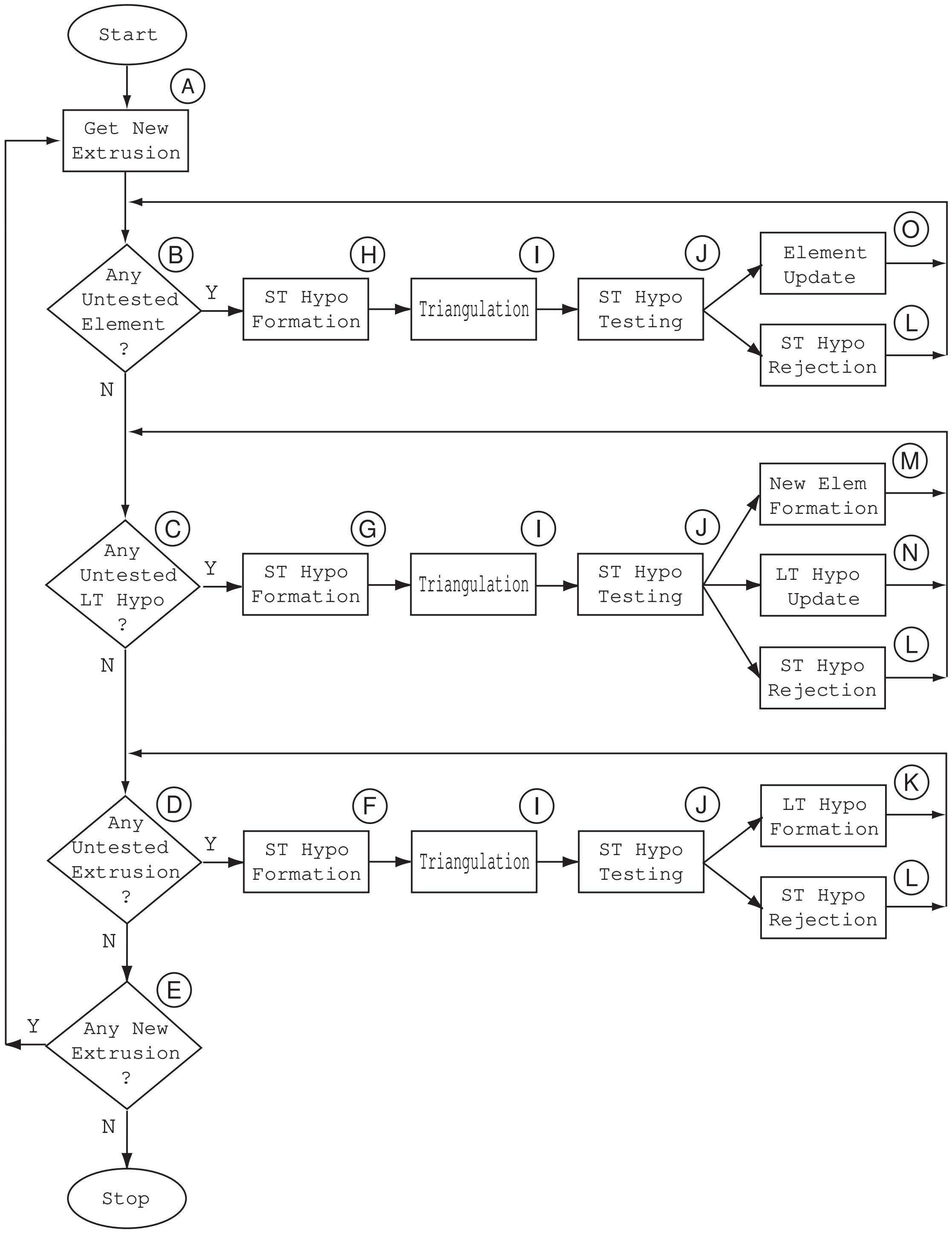


Figure 2.7: Flow diagram of the feature correspondence process. “ST Hypo” stands for short-term hypothesis. “LT Hypo” stands for long-term hypothesis.

I. Perform triangulation on the temporary ST hypothesis. Obtain key measures needed for hypothesis evaluation. (See Chapter 3).

J. Evaluate the validity of the temporary ST hypothesis by computing its posterior probability. (See Chapter 4).

K. If the posterior probability is above the hypothesis rejection threshold, form a new LT hypothesis.

L. If the posterior probability is below the hypothesis rejection threshold, reject the ST hypothesis.

M. If the posterior probability is above the element acceptance threshold, form a new element.

N. If the posterior probability is above the hypothesis rejection threshold, but below the element acceptance threshold, update the current LT hypothesis.

O. If the posterior probability is above the element acceptance threshold, update the current element.

It should be evident that the three basic steps outlined at the beginning of this chapter (hypothesis generation, triangulation, and hypothesis evaluation) are embedded in our correspondence system as three branches to the right of the main flow control. The top branch performs matching with existing elements, the middle branch performs matching with existing hypotheses, and the bottom branch performs matching with existing extrusions. The main loop goes through every input feature and tests each one for possible entry into three branches of match operations.

2.5 Summary

In this chapter, we have accomplished the following tasks:

1. We examined the correspondence problem in the context of a triangulation-based approach. Two common accidental match cases were presented to demonstrate the need for getting multiple supporting features.

2. We described the data structures used in our system, covering three types (vertex, segment, surface), and four classes (feature, extrusion, hypothesis, element). Hypothesis related terminology was explained.

3. We introduced three classes for categorizing incremental correspondence algorithms – linear exact match, linear mixed match, and branching exact match algorithms. The current system belongs to the linear exact match category.

4. We outlined our algorithm for making feature correspondence through a flow diagram. An efficient order of processing was proposed for organizing the matching of new features with existing data structures.

Chapter 3

Triangulation Methods

The triangulation methods are our main tools for solving the dual problems of simultaneous correspondence and reconstruction, which were discussed in Chapter 1. The computation of intersection of extrusions provides us with two important pieces of information suitable for evaluating correspondence hypotheses:

1. an estimate of its 3D structure, and

2. a measure of how well the feature extrusions match in 3D.

In this chapter, we will first describe the formation of temporary short-term hypothesis that occurs immediately before the triangulation step. Then we will describe two multi-image triangulation methods, one for computing the intersection of vertex extrusions, and another for computing the intersection of segment extrusions. In order to highlight the uncertainties involved, the methods are developed in a probabilistic framework. Lastly, we note several special situations under which these methods may fail, and describe ways to get around them.

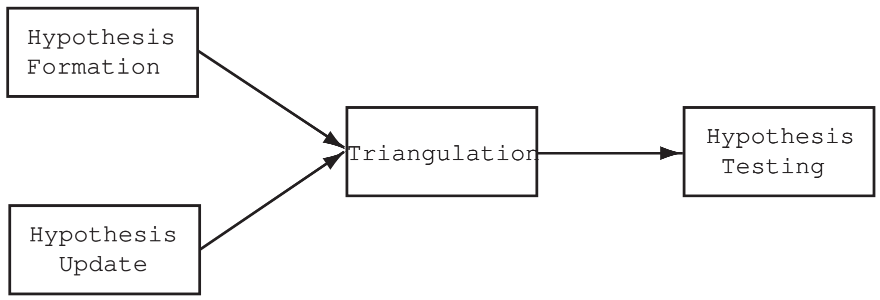


Figure 3.1: Processing pathway of triangulation

3.1 Hypothesis Formation

The incremental process of our reconstruction system is driven by the insertion of new features from new images. To avoid missing any possible 3D structure, plausible correspondence hypotheses must be formed exhaustively for triangulation and hypothesis testing. A diagram depicting the processing pathway before and after the triangulation process is shown in Figure 3.1.

We shall use the abbreviation ST to denote short-term or temporary hypothesis, and LT to denote long-term or tracking hypothesis. Each correspondence hypothesis consists of two or more features of the same type (vertex, segment), but all must come from distinct images. We make a distinction between three types of new hypotheses that are formed immediately before entering the triangulation process:

1. Initial ST hypothesis – formed by combining two features from different images.

2. Combination ST hypothesis – formed by combining a new feature with an existing LT hypothesis or element.

3. Modified LT hypothesis – formed by modifying the feature support of an existing LT hypothesis.

Upon initialization, there is no hypothesis in the system. Features of the same type but from different images are paired up exhaustively to form initial ST hypotheses. They are then fed into the triangulation system one by one. When an initial ST hypothesis passes its first hypothesis test (see Chapter 4), it becomes a LT hypothesis that will be tracked.

As soon as a LT hypothesis is formed, new features entering the system are paired up with the LT hypothesis in addition to old features already in the system, generating new combination ST hypotheses and initial ST hypotheses. The combination ST hypotheses are processed in a similar way as the initial ST hypotheses. However, if the combination ST hypothesis fails its test, the hypothesis is rejected but the original LT hypothesis remains valid.

Likewise, when a LT hypothesis is confirmed as a 3D structural element, new features are paired up with the element to form new combination ST hypotheses.

The feature support of any LT hypothesis is constantly subject to change, even when the focus of processing is on another hypothesis or element. Specifically, there are two types of events which can trigger unexpected feature support modification: (1) confirmation of a competing LT hypothesis, or (2) enforcement of new visibility constraints associated with the formation of a new surface element.

These modified LT hypotheses can enter into the triangulation process directly.

If the modified LT hypothesis fails its hypothesis test, the modified LT hypothesis is rejected.

3.2 Vertex Triangulation

Given the corresponding image positions $\mathbf{m}_i$ of a 3D point $\mathbf{x}$ projected onto a set of images I_i, we would like to estimate the 3D point $\mathbf{x}$. Let us define a vertex extrusion as the 3D ray emanating from a camera center $\mathbf{c}_i$, passing through the image vertex feature $\mathbf{m}_i$, with direction $\mathbf{a}_i$ (see Figure 3.2). The point $\mathbf{x}$ can be estimated as the intersection of relevant vertex extrusions in 3D space.

Due to noise and errors in measurements, vertex extrusions will not intersect precisely at one point. We assume that the closest points on the extrusions to the true point $\mathbf{x}$ are independent and normally distributed around the true point with covariance Λ_i:

$$\mathbf{y}_i = \mathbf{x} + \mathbf{n}_i \qquad \mathbf{n}_i \sim N(0, \Lambda_i) \tag{3.1}$$

where $\mathbf{y}_i$ denotes the point on vertex extrusion i closest to the true point $\mathbf{x}$. The random displacement vector $\mathbf{n}_i$ perturbs extrusion i from the true point. We note that our Gaussian assumption has not been tested or verified.

The conditional probability for the closest point on vertex extrusion i given the 3D point can be written as

$$p(\mathbf{y}_i \mid \mathbf{x}) = p(\mathbf{y}_i - \mathbf{x}) = N(\mathbf{y}_i - \mathbf{x}, \Lambda_i) \tag{3.2}$$

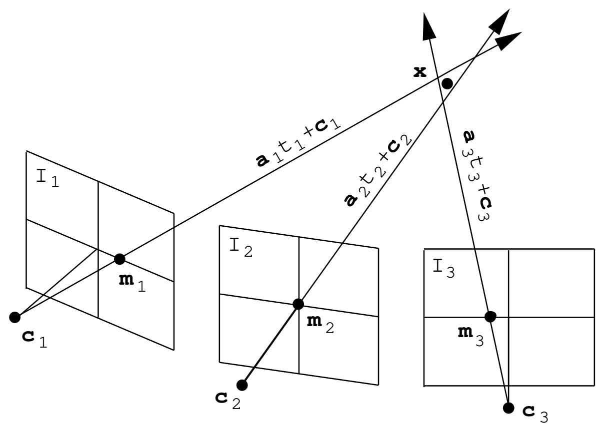


Figure 3.2: Intersection of vertex extrusions

where

$$N(\mathbf{y}_i - \mathbf{x}, \Lambda_i) = \frac{1}{(2\pi)^{3/2}|\Lambda_i|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{y}_i - \mathbf{x})^T \Lambda_i^{-1}(\mathbf{y}_i - \mathbf{x})\right] \tag{3.3}$$

The conditional probability for the set of extrusions intersecting on point $\mathbf{x}$ jointly is the product of individual conditional probability functions:

$$p(\mathbf{Y} \mid \mathbf{x}) = \prod_i p(\mathbf{y}_i \mid \mathbf{x}) = \prod_i N(\mathbf{y}_i - \mathbf{x}, \Lambda_i) \tag{3.4}$$

Since we have no prior knowledge about the possible positions of the true point, estimation of the point will be based on a Maximum Likelihood (ML) approach.

$$\hat{\mathbf{x}}_{ML} = \arg\max_x p(\mathbf{Y} \mid \mathbf{x}) \tag{3.5}$$

Maximization of $p(\mathbf{Y} \mid \mathbf{x})$ is equivalent to maximization of $\ln[p(\mathbf{Y} \mid \mathbf{x})]$, since taking the logarithm does not change the result. Thus, we can estimate $\hat{\mathbf{x}}_{ML}$ as

$$\hat{\mathbf{x}}_{ML} = \arg\max_x \ln[p(\mathbf{Y} \mid \mathbf{x})] \tag{3.6}$$

$$= \arg\max_x \ \ln\left[\prod_i N(\mathbf{y}_i - \mathbf{x}, \Lambda_i)\right] \tag{3.7}$$

$$= \arg\max_x \sum_i \left[-\frac{1}{2}(\mathbf{y}_i - \mathbf{x})^T \Lambda_i^{-1}(\mathbf{y}_i - \mathbf{x})\right] \tag{3.8}$$

We note that the maximization of the summation term is equal to the minimization of the negative of the corresponding term. Thus, the 3D point $\mathbf{x}$ can be estimated by minimizing the sum of weighted squared distances between the extrusions and the point:

$$\hat{\mathbf{x}}_{ML} = \arg\min_x \sum_i \mathbf{d}_i^T(\mathbf{x})\Lambda_i^{-1}\mathbf{d}_i(\mathbf{x}) \tag{3.9}$$

where $\mathbf{d}_i(\mathbf{x}) = \mathbf{y}_i - \mathbf{x}$, is the displacement vector between the true point and the vertex extrusion i.

The closest point $\mathbf{y}_i$ on extrusion ray i to point $\mathbf{x}$ is represented by

$$\mathbf{y}_i = \mathbf{a}_i\, t_i + \mathbf{c}_i \qquad \text{where} \qquad t_i = \mathbf{a}_i^T(\mathbf{x} - \mathbf{c}_i) \tag{3.10}$$

where $\mathbf{a}_i$ is the extrusion vector computed from observed vertex feature $\mathbf{m}_i$, and $\mathbf{c}_i$ is the corresponding camera position.

We now have $\mathbf{d}_i(\mathbf{x})$ as

$$\mathbf{d}_i(\mathbf{x}) \ = \ \mathbf{y}_i - \mathbf{x} \ = \ \mathbf{a}_i\mathbf{a}_i^T(\mathbf{x} - \mathbf{c}_i) + \mathbf{c}_i - \mathbf{x} \tag{3.11}$$

Let us denote the sum of weighted squared distances between the extrusions and the true point as

$$D_V(\mathbf{x}) \equiv \sum_i \mathbf{d}_i^T(\mathbf{x})\Lambda_i^{-1}\mathbf{d}_i(\mathbf{x}) \tag{3.12}$$

We assume that the covariance matrix Λ_i^T is diagonal and may be reduced to constant variance, i.e. $\Lambda_i = \lambda_i I$. The variance λ_i should be small for observed features with high confidence, and large for observed features with low confidence.

Computing the gradient of $D_V(\mathbf{x})$ with respect to $\mathbf{x}$, we get

$$\begin{aligned}
\nabla_{\mathbf{x}}\, D_V(\mathbf{x}) &= \sum_i \nabla_{\mathbf{x}} \left[\mathbf{d}_i(\mathbf{x})^T \Lambda_i^{-1} \mathbf{d}_i(\mathbf{x})\right] && (3.13)\\
&= \sum_i \frac{2}{\lambda_i} \left[\nabla_{\mathbf{x}} \mathbf{d}_i(\mathbf{x})^T\right]\, \mathbf{d}_i(\mathbf{x}) && (3.14)\\
&= \sum_i \frac{2}{\lambda_i} (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I})^T \, \left[\mathbf{a}_i \mathbf{a}_i^T (\mathbf{x} - \mathbf{c}_i) + \mathbf{c}_i - \mathbf{x}\right] && (3.15)
\end{aligned}$$

Setting the gradient to zero leads to

$$\sum_i \frac{1}{\lambda_i} (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I})^T (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I})\, \mathbf{x} \;=\; \sum_i \frac{1}{\lambda_i} (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I})^T (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I}) \mathbf{c}_i \tag{3.16}$$

We note that this is a linear system in the standard form of $\mathbf{Ax} = \mathbf{b}$. Using singular value decomposition (SVD) the left hand side matrix can be decomposed into

$$\sum_i \frac{1}{\lambda_i} (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I})^T (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I}) = \mathbf{UWU}^T \tag{3.17}$$

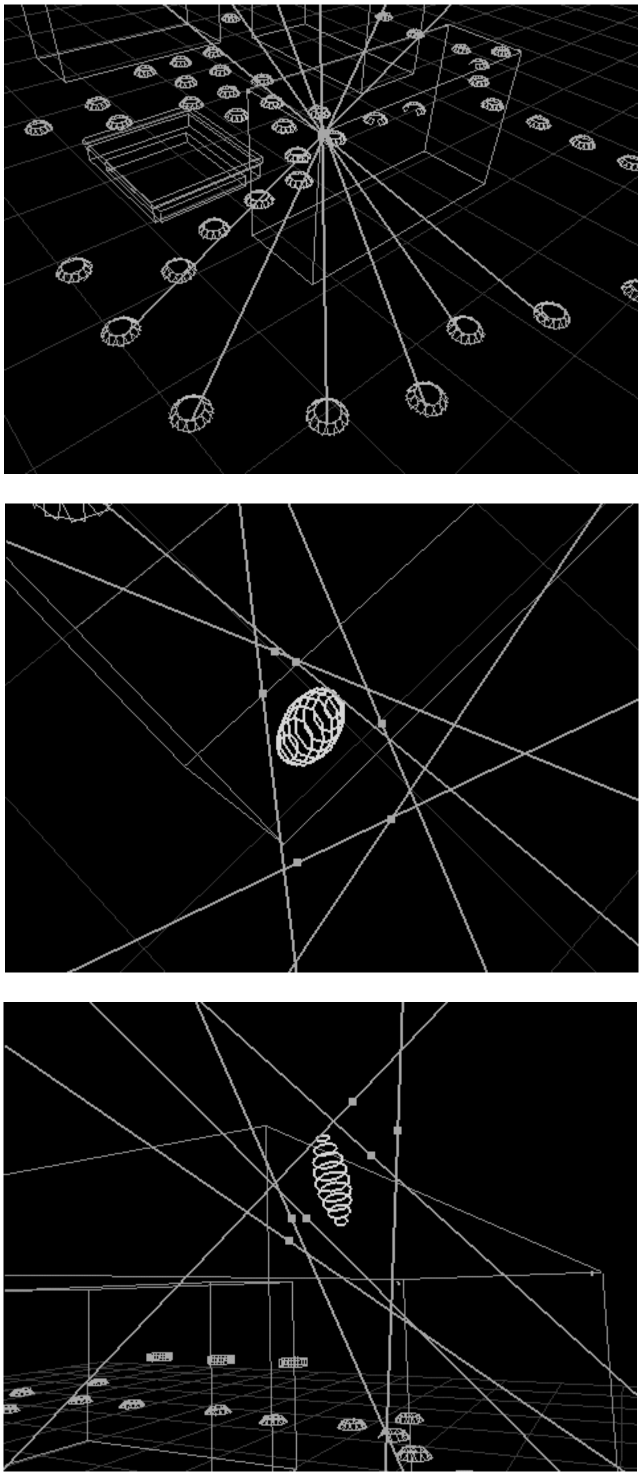
where $\mathbf{U}$ is an orthonormal matrix satisfying $\mathbf{U}^T = \mathbf{U}^{-1}$, and $\mathbf{W}$ is a diagonal matrix containing the singular values [40].

The desired ML estimate of $\mathbf{x}$ is determined to be

$$\hat{\mathbf{x}}_{ML} = \mathbf{UW}^{-1}\mathbf{U}^T \left[\sum_i \frac{1}{\lambda_i} (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I})^T (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I}) \mathbf{c}_i\right] \tag{3.18}$$

The residual of the triangulation process is given by $D_V(\mathbf{x})$ of Equation 3.12.

Figure 3.3 is the visualization of a typical vertex triangulation problem. The principal axes (eigenvectors) and the eigenvalues of the SVD computation are represented by an ellipsoid centered on the estimated point of intersection. Dots on the extrusions represent their nearest points $\mathbf{y}_i$ to the center of intersection.



(a)

(b)

(c)

Figure 3.3: Visualization of a vertex triangulation. (a) A view of the intersection. (b) A close-up top view of the intersection. (c) A close-up side view of the intersection. Dots on the extrusions represent their nearest points to the center of intersection.

3.3 Segment Triangulation

For line segments, the computation is more involved. We describe a three step process to determining the intersection of segment extrusions:

1. Estimate the direction of the solution line
2. Determine a point on the solution line
3. Determine the segment extent on the solution line

Due to the complexities involved, only the first step is formulated in a probabilistic framework.

Our first task is to estimate the 3D line $L : \mathbf{v}t + \mathbf{p}$ at the intersection of segment extrusions. A segment extrusion is the 3D wedge emanating from the camera center $\mathbf{c}_i$, passing through image line segment L_i, and bounded by vectors $\mathbf{v}_{2i-1}$ and $\mathbf{v}_{2i}$ (see Figure 3.4). The normal of the extrusion plane is given by $\mathbf{e}_i = \mathbf{v}_{2i-1} \times \mathbf{v}_{2i}$.

In the ideal case, the direction $\mathbf{v}$ of the intersection line should be perpendicular to the normals ($\mathbf{e}_i$) of the extrusion planes. This gives us the line direction constraint equation

$$\mathbf{e}_i^T \mathbf{v} = 0 \qquad \forall\ i \tag{3.19}$$

A single line direction constraint equation is insufficient to produce a unique $\mathbf{v}$. We must use multiple extrusions from observed segments to estimate the intersection line. In matrix form, the previous constraint equation becomes

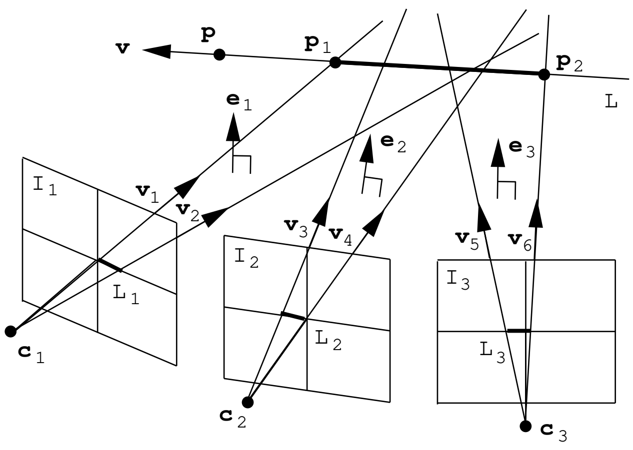


Figure 3.4: Intersection of segment extrusions

$$\begin{bmatrix} e_{x_1} & e_{y_1} & e_{t_1} \\ e_{x_2} & e_{y_2} & e_{t_2} \\ \vdots & \vdots & \vdots \\ e_{x_N} & e_{y_N} & e_{t_N} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = 0$$

$$\mathbf{E} \qquad \mathbf{v} = 0 \tag{3.20}$$

where $\mathbf{E}$ is a matrix whose rows are the normals $\mathbf{e}_i$ of the extrusion wedges.

It should be noted that Equation 3.20 reflects the case when all of our segment measurements are perfectly accurate. In reality, this assumption is undermined by noise in the data and errors in the measurement process. We can model these corrupting factors by introducing a noise factor $\mathbf{n}$ into our equation

$$\mathbf{E} \cdot \mathbf{v} = \mathbf{n} \qquad \mathbf{n} \sim N(0, \Lambda_i) \tag{3.21}$$

where we assume that the noise factor $\mathbf{n}$ is a Gaussian random variable with zero-mean and covariance Λ_n.

From the above equation, we obtain the conditional probability function

$$p(\mathbf{E} \mid \mathbf{v}) = N(\mathbf{E} \cdot \mathbf{v}, \Lambda_n) \tag{3.22}$$

where

$$N(\mathbf{E} \cdot \mathbf{v}, \Lambda_n) = \frac{1}{(2\pi)^{m/2}|\Lambda_n|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{v}^T \cdot \mathbf{E}^T)\Lambda_n^{-1}(\mathbf{E} \cdot \mathbf{v})\right] \tag{3.23}$$

Without any prior information on the true direction of the intersection line, we will estimate the direction vector $\mathbf{v}$ using the Maximum Likelihood (ML) method

$$\hat{\mathbf{v}}_{ML} = \arg\max_{\mathbf{v}} p(\mathbf{E} \mid \mathbf{v}) \tag{3.24}$$

Since taking the logarithm does not change the result of maximization, the maximization of $p(\mathbf{E} \mid \mathbf{v})$ is equivalent to maximization of $\ln[p(\mathbf{E} \mid \mathbf{v})]$. Thus, we can estimate $\hat{\mathbf{v}}_{ML}$ as

$$\begin{aligned} \hat{\mathbf{v}}_{ML} &= \arg\max_{\mathbf{v}} \ln[p(\mathbf{E} \mid \mathbf{v})] && (3.25)\\ &= \arg\max_{\mathbf{v}} \left\{-\frac{1}{2}(\mathbf{v}^T \cdot \mathbf{E}^T)\Lambda_n^{-1}(\mathbf{E} \cdot \mathbf{v})\right\} && (3.26)\\ &= \arg\min_{\mathbf{v}} \left\{\frac{1}{2}(\mathbf{v}^T \cdot \mathbf{E}^T)\Lambda_n^{-1}(\mathbf{E} \cdot \mathbf{v})\right\} && (3.27) \end{aligned}$$

In the last step, the maximization problem is converted to a minimization problem by negation.

An important aspect of the problem we have not considered is that we need to constrain the directional vector as a normalized vector. This is necessary to ensure the solution is not a zero vector. The directional vector $\mathbf{v}$ can be estimated by adding a normalization term $(1 - \mathbf{v}^T\mathbf{v})$ scaled by the Lagrange multiplier λ.

$$\hat{\mathbf{v}}_{ML} = \arg\min_{\mathbf{v}} \left\{\frac{1}{2}(\mathbf{v}^T \cdot \mathbf{E}^T)\Lambda_n^{-1}(\mathbf{E} \cdot \mathbf{v}) + \lambda(1 - \mathbf{v}^T\mathbf{v})\right\} \tag{3.28}$$

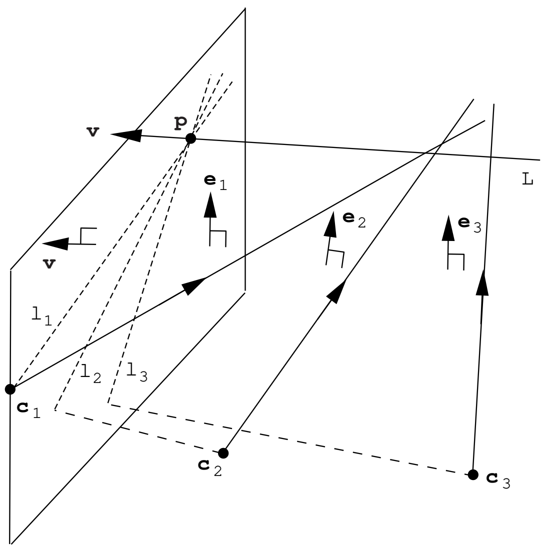


Figure 3.5: Point $\mathbf{p}$ is found to be the intersection of lines produced by the intersection of extrusion planes with the reference plane (same configuration as Figure 3.4).

Taking derivative of the term with respect to $\mathbf{v}$, we get

$$\mathbf{E}^T \Lambda_n^{-1} \mathbf{E} \cdot \mathbf{v} = \lambda \mathbf{v} \tag{3.29}$$

This equation is in the form of a standard eigenvalue problem. The direction vector $\mathbf{v}$ is estimated to be the eigenvector $\mathbf{v}_1$ of matrix $\mathbf{E}^T \Lambda_n^{-1} \mathbf{E}$ with the smallest eigenvalue λ_1. It can be solved by singular value decomposition (SVD) or the QR method [44].

The next step is to estimate a fixed point $\mathbf{p}$ on the 3D line L. First, intersect every extrusion plane $(\mathbf{e}_i, \mathbf{c}_i)$, with the reference plane $(\mathbf{v}, \mathbf{c}_1)$, which has the normal $\mathbf{v}$ as computed above, and contains the point $\mathbf{c}_1$. This produces a set of lines l_i lying on the reference plane (see Figure 3.5). The point $\mathbf{p}$ is estimated to be the 3D point of intersection of lines l_i.

The residual for the segment triangulation process is computed as the sum of squared distances between the bounding rays of segment extrusions and the 3D line L:

$$D_L(\mathbf{v}, \mathbf{p}) = \sum_j \|(\mathbf{v}_j\ s_j + \mathbf{c}_i) - (\mathbf{v}\ t_j + \mathbf{p})\|^2 \tag{3.30}$$

where j indexes the bounding rays, and $i = \lfloor \frac{j+1}{2} \rfloor$ indexes the corresponding cameras.

The closest points on the bounding ray to the solution line are given by $\mathbf{v}_j\ s_j + \mathbf{c}_i$, where

$$s_j = \frac{(\mathbf{v} \times \mathbf{v}_j) \cdot ((\mathbf{c}_i - \mathbf{p}) \times \mathbf{v})}{\|\mathbf{v} \times \mathbf{v}_j\|^2} \tag{3.31}$$

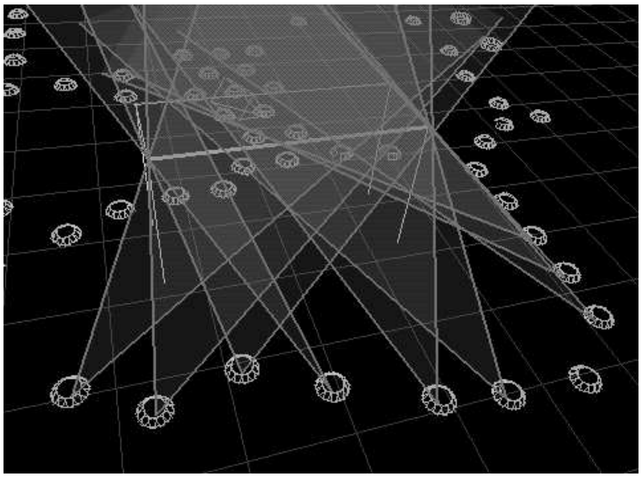
with the constraint of $s_j \geq 0$.

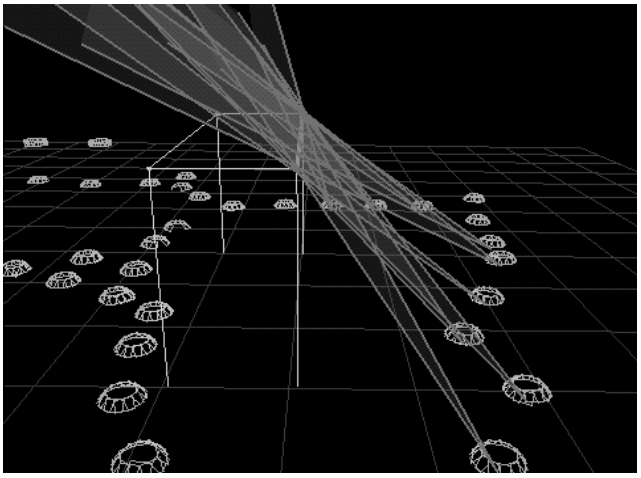
The closest points on the solution line to the bounding rays are given by $\mathbf{v}\ t_j + \mathbf{p}$, where

$$t_j = \frac{(\mathbf{v} \times \mathbf{v}_j) \cdot ((\mathbf{c}_i - \mathbf{p}) \times \mathbf{v}_j)}{\|\mathbf{v} \times \mathbf{v}_j\|^2} \tag{3.32}$$

Once the 3D line L is defined, the solution line segment $(\mathbf{p}_1, \mathbf{p}_2)$ can be estimated as follows. First, determine the union of line extrusions, and make sure the union is continuous. Then intersect the two rays bounding the union with the 3D line L. The endpoints $\mathbf{p}_1$ and $\mathbf{p}_2$ are given by $\mathbf{p}_j = \mathbf{v}\ t_j + \mathbf{p}$, where t_j are computed by Equation 3.32 with $\mathbf{v}_j$ as the bounding rays.

The triangulation of a horizontal line segment is visualized in Figures 3.6(a)(b). The triangulation of a vertical line segment is visualized in Figure 3.6(c).


(a)


(b)

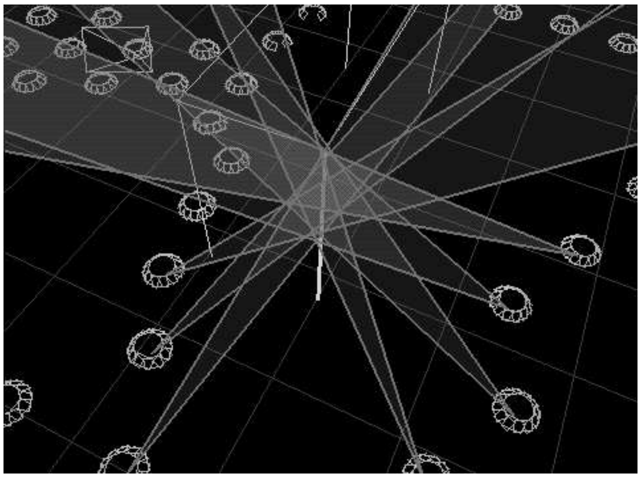

(c)

Figure 3.6: Visualization of two segment triangulations: (a) View of a horizontal segment triangulation. (b) A side view of the same triangulation. (c) View of a vertical segment triangulation.

3.4 Special Situations

While the triangulation methods just presented appear to be quite general, there are several special situations which we must be aware of. These situations are listed here and illustrated in Figure 3.7. These situations can all result in numerical instabilities.

- Case I, Figure 3.7(a): vertex extrusions originating from the same camera position with a very small angular separation θ. These extrusions tend to be the same vertex observed in different images, and will result in unstable intersection. Solution: merge the vertex extrusions.

- Case II, Figure 3.7(b): vertex extrusions originating from different camera positions, forming a highly acute intersection angle θ. The point of intersection formed in this case is unstable, as a small change in θ can vary the result dramatically. Solution: invalidate the correspondence.

- Case III, Figure 3.7(c): segment extrusions originating from the same camera position with a very small angular separation θ between their normals. These extrusions tend to be the same line segment observed in different images, and will result in unstable intersection. Solution: merge the segment extrusions.

- Case IV, Figure 3.7(d): segment extrusions originating from different camera positions, forming a highly acute angle θ with solution line. The segment of intersection formed in this case is unstable, as a small change in θ can vary the result dramatically. Solution: invalidate the correspondence.

In our system, cases I and III are detected and resolved in the initial processing stage, before the system enters the hypothesis formation and triangulation stages. Cases II and IV are detected and resolved during the triangulation stage, with II before the intersection of extrusions, and IV after the intersection.

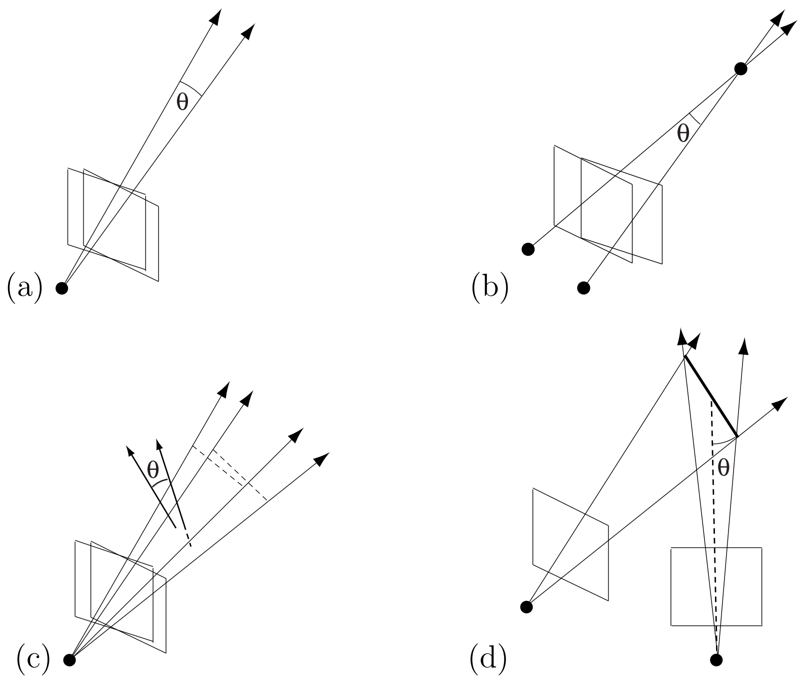


Figure 3.7: Special situations (very small θ): (a) Case I – unstable vertex intersection. (b) Case II – unstable vertex intersection. (c) Case III – unstable segment intersection. (d) Case IV – unstable segment intersection.

3.5 Summary

In this chapter, we have accomplished the following tasks:

1. We outlined the hypothesis formation process that precedes the triangulation step. Three types of new correspondence hypotheses are noted, and their different possible outcomes characterized.

2. We derived the multi-image triangulation method for computing the intersection of vertex extrusions through probabilistic modeling and Maximum Likelihood estimation. The solution is determined by solving a standard linear system.

3. We formulated the multi-image triangulation method for computing the intersection of segment extrusions. The solution involves three separate steps: 1) estimating the direction of the solution line, 2) determining a point on the solution line,

and 3) determining the extent of the segment on the solution line.

4. We noted four special situations under which the triangulation methods may fail. They are all due to numerical instabilities associated with small angle intersections. Ways to get around the problems have been recommended.

Chapter 4

Hypothesis Processing

The selection of probable correspondence hypotheses plays an important role in our system. In a large-scale reconstruction project, the number of hypotheses generated will be massive, yet most of them are invalid and inconsistent with one another. It is practical to prune as many improbable hypotheses as quickly as possible in order not to over-burden the system. At the same time, we must be careful not to eliminate any hypothesis that could become a 3D structural element.

To meet these two conflicting needs, we have developed a decision system based on probabilistic modeling of two key measures of a hypothesis' fitness. The first measure is a function of the triangulation residual, while the second is a function on the number of supporting features. Each of these measures can provide a probabilistic estimate on the validity of the hypothesis.

These two measures are integrated into a single hypothesis posterior probability. Based on this probability, a decision will be made for each hypothesis. In general, there are three possible outcomes: confirmation to a 3D structural element, hypothesis rejection, or update and wait for more information (see Figure 4.1).

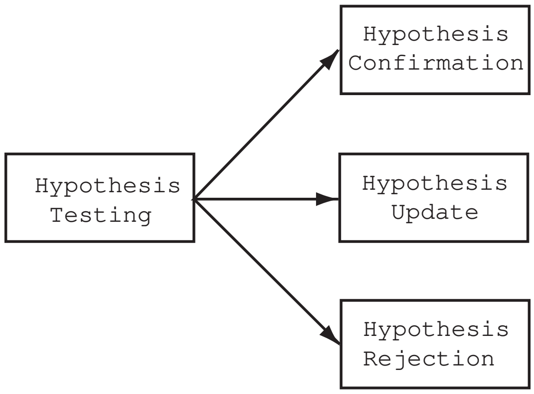


Figure 4.1: Possible outcomes of hypothesis testing

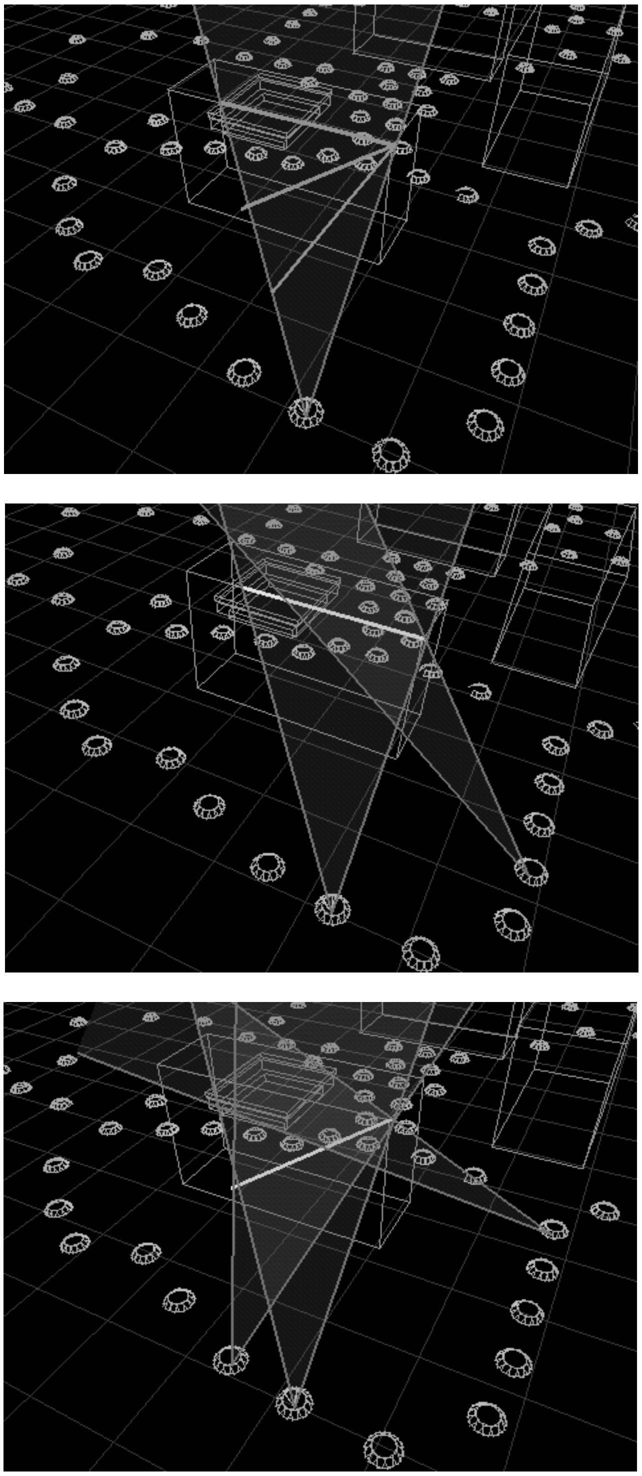
4.1 Hypothesis Competition

When hypotheses share common supporting features, they are in competition. Under ideal situation, each feature should be linked with only one hypothesis, since each feature can be the projection of a single 3D structure.

A typical instance of hypothesis competition is shown in Figure 4.2. Three competing segment hypotheses are associated with a segment extrusion. Two of the hypotheses along with their supporting extrusions are illustrated in Figures 4.2 (b) and (c). Only the hypothesis in (b) is correct while the other two are incorrect.

We note that the hypothesis in Figure 4.2(c) has more supporting features/extrusions than the correct (b) hypothesis at the time when the snapshot was taken. However, over time, the correct (b) hypothesis will collect many more supporting extrusions, as appears in Figure 3.6(a).

In determining which hypothesis is the correct one, we will use two key measures



(a)

(b)

(c)

Figure 4.2: Visualization of competing segment hypotheses. (a) Three competing segment hypotheses associated with a segment extrusion. (b) Extrusions associated with the correct hypotheses. (c) Extrusions associated with a competing hypothesis.

from the triangulation process. The first measure is the triangulation residual, which correlates with how well the extrusions intersect in 3D. The second measure is the number of supporting features, which reflects the number of consistent observations. From these two measures, we can compute the data consistency probability and the non-accidentalness probability, as described in the following sections.

4.2 Data Consistency Probability

The most basic requirement for a valid hypothesis is that the intersection error should be small. This requirement can be embedded in a data consistency probability function for filtering out hypotheses with large intersection error.

Given triangulation error D, we can compute the posterior probability of a correspondence hypothesis given D. Let H_1 denote the hypothesis that the image features do arise from a single 3D object (vertex or line segment), and H_0 denote the null hypothesis. Their prior probabilities will be represented by $p(H_1) = \varphi$ and $p(H_0) = 1 - \varphi$, respectively.

The data consistency probability $p(H_1 \mid D)$ can be derived by applying Bayes' rule:

$$p(H_1 \mid D) = \frac{p(D \mid H_1)p(H_1)}{p(D \mid H_0)p(H_0) + p(D \mid H_1)p(H_1)} \tag{4.1}$$

The denominator is the normalizing factor from Bayes' rule over the binary sample space, $\{H_0, H_1\}$.

In our computation, we will use average intersection error, or absolute intersection residual divided by the number of supporting features. Averaging is necessary because the absolute intersection residual almost always increases with the number of supporting features. To remove the bias for hypotheses with few supporting features, averaging is performed.

The correspondence hypothesis likelihood $p(D \mid H_1)$ is a Gaussian function of the intersection error of feature extrusions in 3D space:

$$p(D \mid H_1) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-D}{2\sigma^2}\right) \tag{4.2}$$

where D is the average intersection error as computed by D_V in Equation 3.12 for vertices, or D_L in Equation 3.30 for line segments, and divided by the number of supporting features $\mathcal{F}$. The standard deviation σ is replaced by σ_v for vertices and σ_l for line segments, respectively.

The null hypothesis likelihood $p(D \mid H_0)$ is assumed to be a uniform distribution over the space of all possible intersection errors $\mathcal{E}$, or $p(D \mid H_0) = 1/\mathcal{E}$.

Now, Equation 4.1 can be written as

$$p(H_1 \mid D) = \frac{1}{1 + \left(\frac{1-\varphi}{\varphi}\right)\left(\frac{\sqrt{2\pi}\sigma}{\mathcal{E}}\right)\exp^{-1}\left(\frac{-D}{2\sigma^2}\right)} \tag{4.3}$$

The data consistency probability $p(H_1 \mid D)$ is an inverse sigmoid function of the triangulation error distance (see Figure 4.3). When D is close to 0, the correspondence hypothesis likelihood $p(D \mid H_1)$ in the denominator of Equation 4.1 is much bigger than the null hypothesis likelihood $p(D \mid H_0)$, resulting in a high overall data consistency

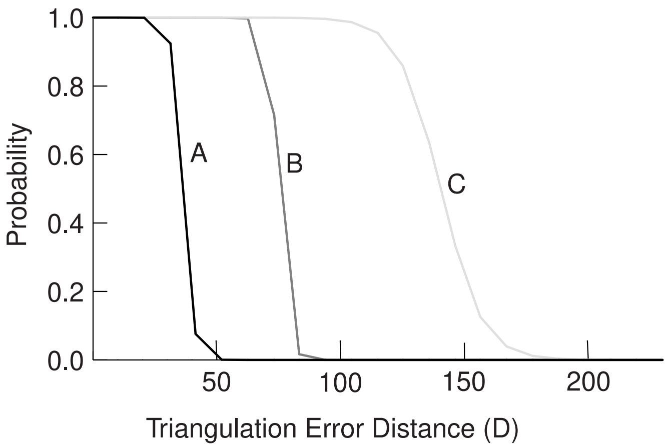


Figure 4.3: Data consistency probability function $p(H_1 \mid D)$. Line A: $\sigma = 1.0$, $1/\mathcal{E} = 1.0^{-8}$, $\varphi = 0.5$; Line B: $\sigma = 1.0$, $1/\mathcal{E} = 1.0^{-16}$, $\varphi = 0.5$; Line C: $\sigma = 2.0$, $1/\mathcal{E} = 1.0^{-8}$, $\varphi = 0.5$.

probability $p(H_1 \mid D)$. However, as D gets larger, $p(D \mid H_0)$ eventually dominates $p(D \mid H_1)$ in the denominator, at which point $p(H_1 \mid D)$ quickly drops to zero. The sharp drop-off in the data consistency probability causes it to behave like a traditional threshold function.

4.3 Non-Accidentalness Probability

Intuitively we know that as the number of intersecting features increases, the probability that the hypothesis is caused by extrusions intersecting *non-accidentally* should increase. If we can determine this probability function mathematically, we will have another way of filtering out accidental hypotheses from non-accidental ones.

Given the number of supporting features $\mathcal{F}$, we can compute the posterior probability of a correspondence hypothesis given $\mathcal{F}$. To derive the non-accidentalness probability $p(H_1 \mid \mathcal{F})$, we model each feature extrusion intersection event as the outcome of a Bernoulli trial. Let p_1 denote the probability of extrusion intersection for features corresponding to the same 3D structure. Let p_0 denote the probability of random intersection for features not corresponding to the same 3D structure.

After a series of n Bernoulli trials, each with the same probability of intersections, the probability of $\mathcal{F}$ supporting features intersecting together corresponding to the same 3D structure is given by the shifted binomial function:

$$p(k \mid H_0) = \binom{n}{\mathcal{F}-1} {p_0}^{\mathcal{F}-1}(1-p_0)^{n-(\mathcal{F}-1)} \qquad \mathcal{F} = 1,\ 2,\ 3,\ \ldots \tag{4.4}$$

where $\mathcal{F}-1$ represents a right shift by one to account for the fact that the minimum number of supporting features for a correspondence hypothesis is 2.

Under the same conditions, the probability of $\mathcal{F}$ random features intersecting together is given by the same shifted binomial function:

$$p(k \mid H_1) = \binom{n}{\mathcal{F}-1} {p_1}^{\mathcal{F}-1}(1-p_1)^{n-(\mathcal{F}-1)} \qquad \mathcal{F} = 1,\ 2,\ 3,\ \ldots \tag{4.5}$$

except for $p_1 \neq p_0$, since accidental intersections are unrelated to correct intersections.

The non-accidentalness probability can be determined by applying Bayes' rule:

$$p(H_1 \mid \mathcal{F}) = \frac{p(\mathcal{F} \mid H_1)p(H_1)}{p(\mathcal{F} \mid H_0)p(H_0) + p(\mathcal{F} \mid H_1)p(H_1)} \tag{4.6}$$

$$= \frac{\varphi\, {p_1}^{\mathcal{F}-1}(1-p_1)^{n-(\mathcal{F}-1)}}{(1-\varphi)\, {p_0}^{\mathcal{F}-1}(1-p_0)^{n-(\mathcal{F}-1)} + \varphi\, {p_1}^{\mathcal{F}-1}(1-p_1)^{n-(\mathcal{F}-1)}} \tag{4.7}$$

$$= \frac{1}{1 + \left(\frac{1-\varphi}{\varphi}\right)\left(\frac{p_0}{p_1}\right)^{\mathcal{F}-1}\left(\frac{1-p_0}{1-p_1}\right)^{n-(\mathcal{F}-1)}} \tag{4.8}$$

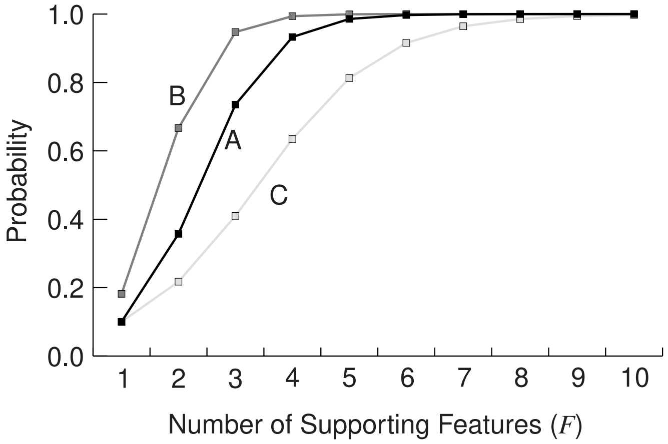


Figure 4.4: Non-accidental match probability function $p(H_1 \mid \mathcal{F})$. Line A: $p_o = 0.1,\ p_1 = 0.5,\ \varphi = 0.1$; Line B: $p_o = 0.1,\ p_1 = 0.9,\ \varphi = 0.2$; Line C: $p_o = 0.2,\ p_1 = 0.5,\ \varphi = 0.1$.

As shown in Figure 4.4, the non-accidentalness probability function $p(H_1 \mid \mathcal{F})$ is a saturation function on the number of supporting features. When $\mathcal{F}$ gets large, $p(H_1 \mid \mathcal{F})$ approaches one. Increasing p_0 shifts the upper part of curve to the right, since it means accidental matches are more likely, so more feature support are required to show that the hypothesis is not an accident. Increasing p_1 or φ shifts the upper part of the curve to the left, for the opposite reason. We assume that $n = \mathcal{F} - 1$ in all of these cases. The results for $\mathcal{F} = 1$ is not meaningful and therefore should be ignored.

4.4 Hypothesis Posterior Probability

Now we can compute a single hypothesis posterior probability based on both the average triangulation error D, and the number of supporting features $\mathcal{F}$. From Bayes' rule, the posterior probability can be written as:

$$p(H_1 \mid D,\ \mathcal{F}) = \frac{p(H_1, \mathcal{F})\, p(D \mid H_1, \mathcal{F})}{p(D, \mathcal{F})} \tag{4.9}$$

Assuming independence between D and $\mathcal{F}$, we get

$$p(H_1 \mid D,\ \mathcal{F}) = \frac{p(H_1 \mid \mathcal{F})\, p(\mathcal{F})\, p(D \mid H_1)}{p(\mathcal{F})\, p(D)} \tag{4.10}$$

Multiplying the top and bottom by $p(H_1)$, yields

$$p(H_1 \mid D,\ \mathcal{F}) = \frac{p(H_1 \mid F)\, p(D, H_1)}{p(D)\, p(H_1)} \tag{4.11}$$

After decomposing $p(D, H_1)$, we obtain the following equation

$$p(H_1 \mid D,\ \mathcal{F}) = \frac{p(H_1 \mid D)\, p(H_1 \mid \mathcal{F})}{p(H_1)} \tag{4.12}$$

This equation suggests that we can independently compute the posterior probability of correspondence hypothesis H_1 given triangulation error D, and the number of supporting features $\mathcal{F}$. We recognize that the first term in the numerator is simply the data consistency probability (Equation 4.3), while the second term is the non-accidentalness probability (Equation 4.8). Both have been derived already. The denominator of Equation 4.12 is simply the prior for the correspondence hypothesis H_1.

4.5 Hypothesis Confirmation

Upon the formation or modification of a correspondence hypothesis, we test to see if

$$p(H_1 \mid D,\ \mathcal{F}) > p_c \tag{4.13}$$

where p_c is a threshold controlling the confirmation of correspondence hypotheses. An increase in the posterior probability of a hypothesis can result from two situations: (1) the addition of a consistent feature support, or (2) the removal of an inconsistent feature support which decreases triangulation error. The first situation is far more likely than the second one.

If the condition of Equation 4.13 is true, the correspondence hypothesis is considered to be highly reliable, and it is promoted to an element. To ensure consistency between features, hypotheses, and elements, the following sequence of processing is initiated:

1. Commit all supporting features of the confirmed hypothesis to the new element. Sever all links between these features and other competing hypotheses.
2. Perform triangulation on all competing hypotheses with modified feature support.
3. Check for possible confirmation, update, or rejection of modified hypotheses via the conditions of Equations 4.13, 4.14, and 4.15.
4. If the hypothesis is of segment type, check to see if a new surface hypothesis or element should be formed. This could result in the reduction of feature support for hypotheses that violate the visibility constraint of the new surface. For further details on surface processing, see Chapter 5.

A flow diagram depicting the above processing sequence is presented in Figure 4.5.

When a hypothesis is confirmed as an element, most of its competing hypotheses will be affected in a negative way due to a reduction in their feature support. Formation of new surface elements may result in further reduction of feature support to other hypotheses. Thus, it should be clear that the confirmation of correspondence hypotheses

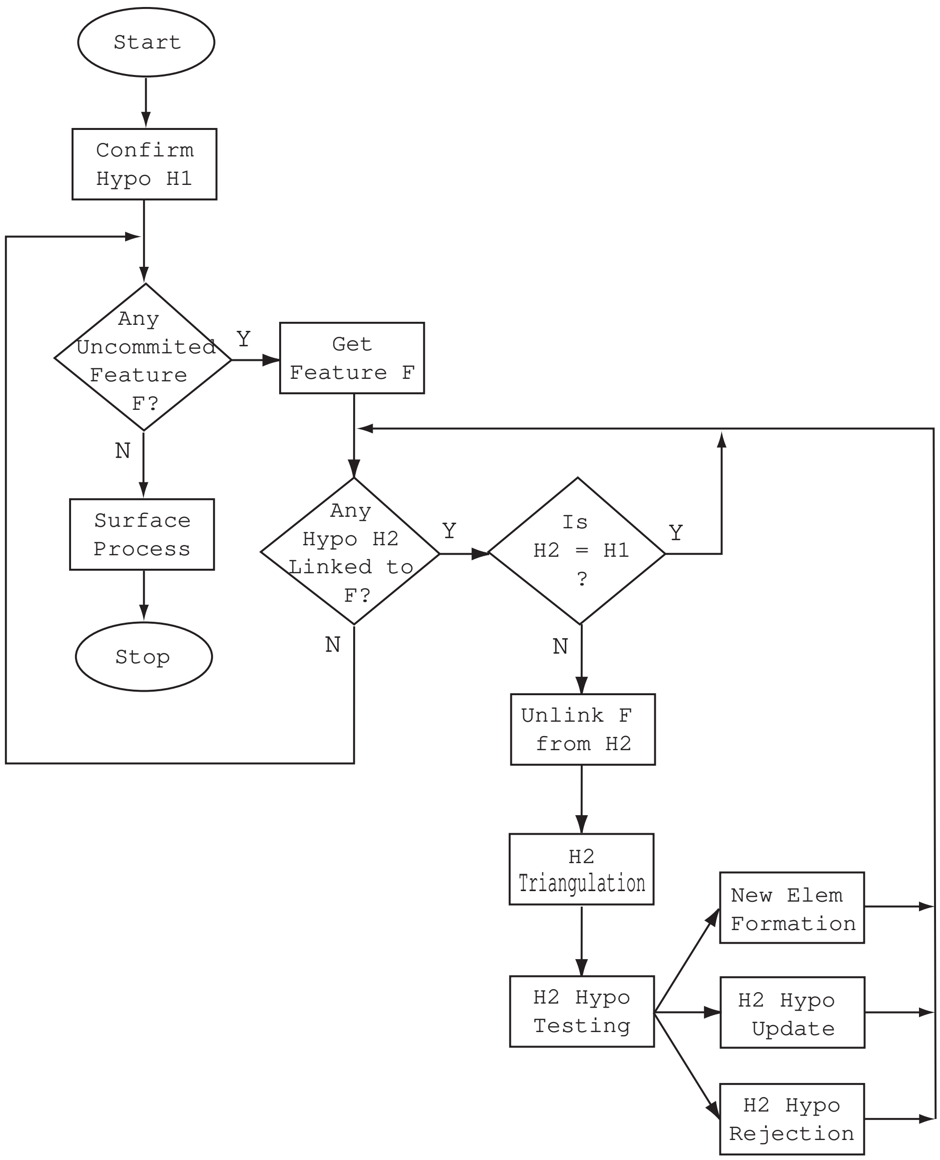


Figure 4.5: Flow diagram of the hypothesis confirmation process

is crucial for the elimination of invalid hypotheses in the system.

4.6 Hypothesis Rejection

The posterior probability of a hypothesis may be decreased for two possible reasons: (1) a reduction in feature support caused by the confirmation of a competing hypothesis, or (2) an increase in triangulation error due to changes in feature support.

The criterion for hypothesis rejection is

$$p(H_1 \mid D,\ \mathcal{F}) < p_r \tag{4.14}$$

where $p_r < p_c$.

Once a hypothesis is rejected, all of the links with its supporting features will be severed. The links between these features and other hypotheses remain unchanged. Thus, unlike hypothesis confirmation, the rejection of a hypothesis does not affect any other hypothesis.

4.7 Hypothesis Update

If the hypothesis posterior probability is greater than the rejection threshold p_r but less than the confirmation threshold p_c, we cannot make an immediate decision on the hypothesis. Instead, we have to wait for more evidence. If a new extrusion is consistent with the current hypothesis, we should update the hypothesis to include the extrusion.

In this case, the hypothesis posterior probability falls between the two decision thresholds:

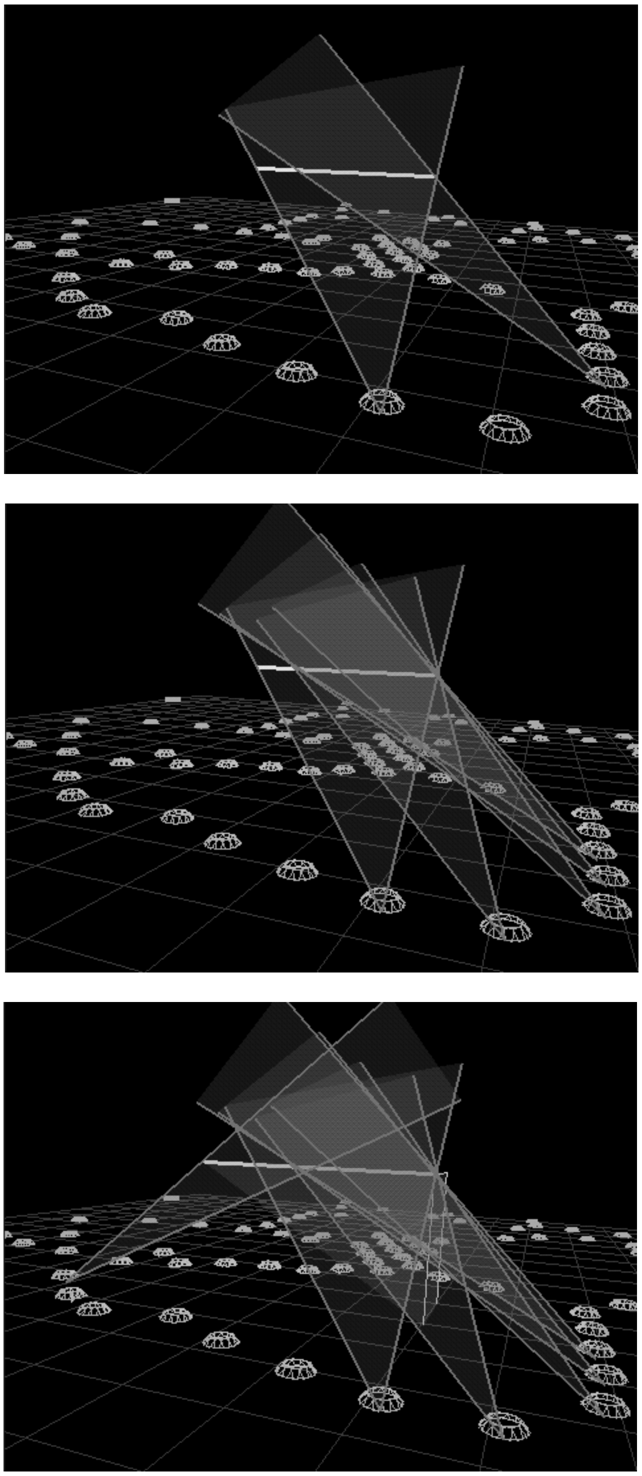
$$p_r < p(H_1 \mid D,\ \mathcal{F}) < p_c \tag{4.15}$$

Specific algorithm for the update of correspondence hypothesis follows the presentation of incremental processing in Chapter 2. Under the Linear Exact Match (LEM) correspondence scheme, which is the existing implementation, each hypothesis with one or more potential matches in an input image is matched up only with the best matching (nearest neighbor) feature extrusion. The updated hypothesis ignores all other features and moves on to the next image.

Under the Linear Mixed Match (LMM) correspondence scheme, each hypothesis is updated by an weighted average of potential matches. The weighting function may be biased toward closer features, or it may be uniform for all features. Under the Branching Exact Match (BEM) correspondence scheme, each hypothesis is split into multiple hypotheses, each one being composed of the original hypothesis, plus one of the potential matching features in the new image. Over time, this strategy generates hypotheses that are related by a tree structure. See Figure 2.6.

Clearly, the choice of the specific update algorithm will have a significant impact on the structure and the performance of the overall reconstruction system. It is not clear which algorithm is best suited for which kind of input. This aspect of research is not addressed in the current thesis.

In Figure 4.6, we follow the development of a successful segment hypothesis. Most correspondence hypotheses are not so successful, and are either eliminated or stay undecided in the system to the end of the reconstruction process.



(a)

(b)

(c)

Figure 4.6: The development of a segment hypothesis. (a) Formation of the hypothesis from two segment extrusions. (b) The hypothesis grows with the addition of new extrusions. (c) The hypothesis reaches its full extent and is confirmed as an element.

4.8 Summary

In this chapter, we have accomplished the following tasks:

1. We specified the data consistency probability function which is a function of the average triangulation residual of a correspondence hypothesis. The probability function has a sharp cut-off which makes it to behave like a threshold.

2. We specified the non-accidentalness probability function which is a function of the number of supporting features of a correspondence hypothesis. The probability function is similar to a saturation function.

3. We defined how decision is made for a correspondence hypothesis based on its posterior hypothesis probability. Processing involved in each of the three main outcomes – hypothesis confirmation, update, and rejection, are considered.

4. We described the chain of events that occurs after the confirmation of a correspondence hypothesis. This includes: commitment of supporting features, update of competing hypotheses, and surface processing.

Chapter 5

Surface Computation

Surfaces in the real world display a wide range of properties, such as roughness, curvature, fuzziness, transparency, reflectivity, and so on. The difficulty of modeling these properties comprehensively should be obvious. However, for many man-made structures, we often can take advantage of two simplifying assumptions about the surfaces:

1. Flatness – allows the surfaces to be modeled by piece-wise planar polygons.

2. Opaqueness – assumes that everything behind a surface is invisible.

These assumptions are incorporated into our computation of surfaces in two major areas: surface parameter estimation, and enforcement of the visibility constraints of confirmed surfaces. To describe a surface efficiently, its parameters are needed. By assuming that all surfaces of interest are relatively planar, the parameter estimation task is greatly simplified. The second task assumes that all surfaces of interest are opaque. The effect is that feature extrusions should not penetrate confirmed surfaces in forming correspondences. Thus, recovered surfaces can be used to detect and eliminate invalid correspondence hypotheses.

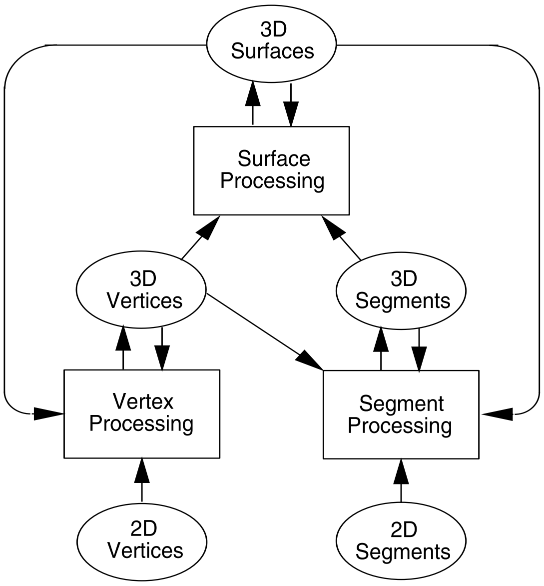


Figure 5.1: Flow of information in our concurrent multi-level reconstruction system

5.1 Concurrent Multi-Level Processing

With the addition of a surface-level processing stage after low-level feature processing stages, our reconstruction system becomes a multi-level system. The main implication of a multi-level system is that information processing no longer happens purely on a local level. Higher-levels of the system, which can aggregate information from across multiple low-level sources, tend to have a more global view of the input.

Non-local or global information can provide important constraints on the processing of low-level information. Low-level processes, by default, face many local ambiguities in making decisions. Without additional information, these low-level processes cannot commit to any decision too soon, without making many mistakes. This is also known as

Marr's *Principle of Least Commitment* [36]. However, the additional constraints from higher-levels can play an important role in these decision making situations. Given their non-local perspective, these constraints can quickly reduce local ambiguities if applied to the right situations.

Ssurface-level information is valuable for the visibility constraints it can provide to low-level correspondence processes [10]. By aggregating information from multiple confirmed vertices and segments, surfaces define an extended area where visual information is blocked. Knowing that confirmed surfaces can quickly eliminate many invalid correspondence hypotheses, it is therefore desirable to generate surfaces as fast as possible, so they can intervene in the processing of correspondence hypotheses. The result is a **concurrent multi-level** reconstruction system. The flow of information in our system is shown in Figure 5.1. High-level descriptions (objects, parts) are not addressed in our current system.

5.2 Surface Algorithm Outline

As mentioned before, there are two major tasks involved in surface computation – surface hypothesis processing, and the enforcement of surface visibility constraints. The processing steps involved in both tasks are outlined in Figure 5.2. Each of the following labeled steps correspond to an identically labeled box in the flow diagram:

A. Confirm segment hypothesis. Commit all supporting extrusions of the new element. Remove links between these features with other hypotheses. Check for rejection of competing hypotheses.

B. Check to see if there is any other segment element linked to the present segment at either one of the two endpoints. If none, stop.

C. Form a new surface hypothesis S1 based on the present segment element and the connecting segment element.

D. Check to see if there is any other surface hypothesis that share the same connecting segment element, thus overlapping with S1.

E. If there are other surface hypotheses sharing the same connecting segment, check to see if the surface hypotheses can be merged by comparing their plane normals.

F. Merge compatible surface hypotheses. Estimate the parameters of the merged surface hypothesis S2, and update the links of supporting segment elements.

G. Check for closure of the merged surface hypothesis S2 by following the segment elements. For vertical surfaces, closure at the ground level is not required.

H. If surface S2 is not closed, check to see if there is another segment element linked to the present segment at the other endpoint. If none, stop.

I. If surface S2 is closed, confirm merged surface hypothesis S2 as a new surface element.

J. Check to see if there is any vertex or segment hypothesis H that has not been tested for violation of the visibility constraints of surface S2.

K. Check to see if there is any vertex or segment extrusion E linked to hypothesis H that has not been tested for violation of the visibility constraints of surface S2.

L. Check to see if extrusion E penetrates the surface S2 in forming hypothesis H. If it does, it's a violation of the visibility constraint of surface S2.

M. Remove violating extrusion E from hypothesis H.

N. Update hypothesis H, and check to see if it should be rejected for a lack of sufficient number of feature support.

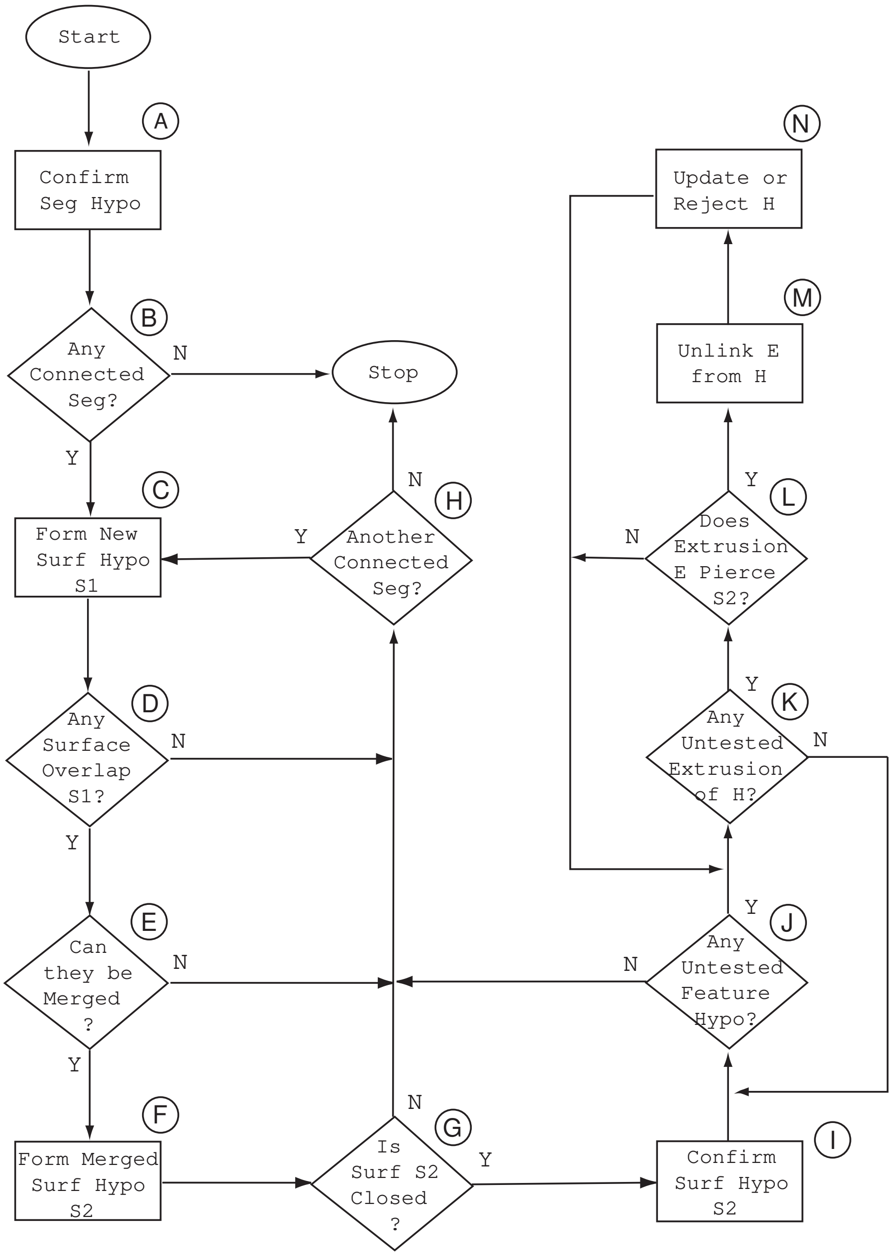


Figure 5.2: Flow diagram of surface processing

Surface hypothesis processing is activated when a segment hypothesis is confirmed and there are other confirmed segments connected to it through the endpoints. The main steps involve hypothesis formation, hypothesis merger, and hypothesis confirmation. They form the left and bottom chains of processing in Figure 5.2.

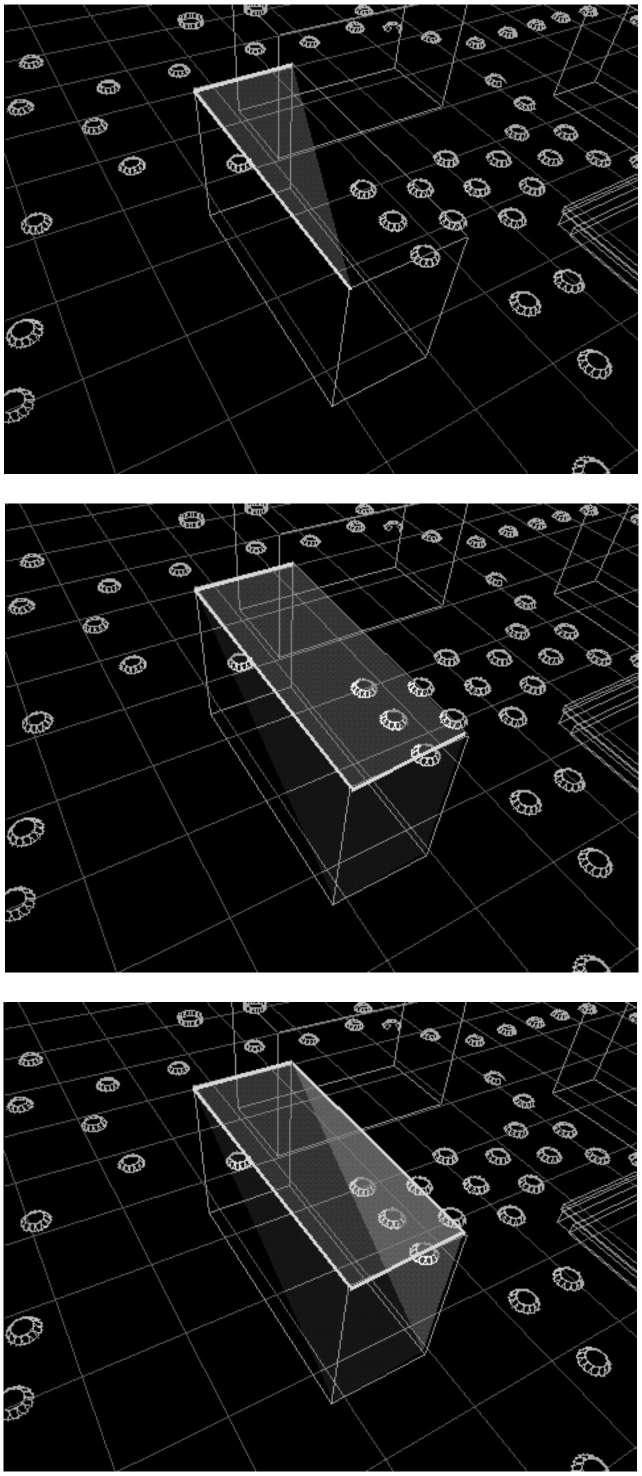
Enforcement of the visibility constraints of confirmed surfaces happens after the confirmation of a new surface element. When that occurs, every extrusion of every existing vertex and segment hypotheses is checked for violation of the new visibility constraints. The right branch in Figure 5.2 corresponds to this process.

A visualization of surface hypothesis processing is shown in Figure 5.3. In (a), a triangular surface hypothesis is formed from two connected segments. Note that all surface hypotheses begin with a triangular shape. In (b), the surface hypothesis attains a rectangular shape with the merger of two triangular surface hypotheses, both sharing a common segment element. However, the hypothesis is not a closed polygon. In (c), the surface hypothesis finally becomes a confirmed surface element when all bounding segments are confirmed.

5.3 Planar Surface Estimation

Based on our assumption that all surfaces of interest can be modeled as piece-wise planar polygons, in this section, we describe how the surface parameters can be estimated.

Let $\mathbf{x}_i$ denote the set of endpoints from segments that define a planar polygon. We can estimate the parameters of the plane, its normal $\mathbf{n}$, and a point $\mathbf{p}$ on the plane, by finding the best-fitting plane through the set of points $\mathbf{x}_i$. See Figure 5.4.



(a) (b) (c)

Figure 5.3: Development of a surface hypothesis. (a) Formation of a surface hypothesis from two connected segments. (b) The hypothesis grows with additional segments. (c) The hypothesis is confirmed when all bounding segments become confirmed.

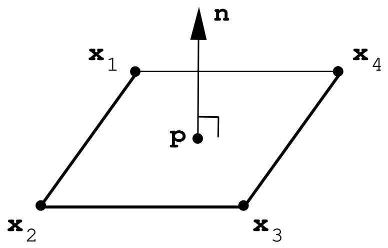


Figure 5.4: Estimating surface parameters $\mathbf{n}$ and $\mathbf{p}$ from segment endpoints $\mathbf{x}_i$.

Ideally, the displacement vector $\mathbf{x}_i - \mathbf{p}$ should be perpendicular to the plane normal $\mathbf{n}$. This constraint is represented by

$$\mathbf{n}^T(\mathbf{x}_i - \mathbf{p}) = 0 \qquad \forall\, i \tag{5.1}$$

However, due to noise and errors in measurement, the contributing segment endpoints $\mathbf{x}_i$ will not all lie on a single plane. We assume that the deviation between the measured segment endpoints and the underlying plane can be modeled by the Gaussian noise $\mathbf{w}_i$ with covariance Λ_i. The new constraint is represented as

$$\mathbf{n}^T(\mathbf{x}_i - \mathbf{p}) = \mathbf{w}_i \qquad \mathbf{w}_i \sim N(0, \Lambda_i) \tag{5.2}$$

The conditional probability for the segment endpoints given the surface parameters $\mathbf{n}$ and $\mathbf{p}$ is

$$p(\mathbf{x}_i \mid \mathbf{n}, \mathbf{p}) = N(\mathbf{n}^T(\mathbf{x}_i - \mathbf{p}), \Lambda_i) \tag{5.3}$$

where

$$N(\mathbf{n}^T(\mathbf{x}_i - \mathbf{p}), \Lambda_i) = \frac{1}{(2\pi)^{3/2}|\Lambda_i|^{1/2}} \exp\left[-\frac{1}{2}\mathbf{n}^T(\mathbf{x}_i - \mathbf{p})\Lambda_i^{-1}(\mathbf{x}_i - \mathbf{p})^T\mathbf{n}\right] \tag{5.4}$$

The joint conditional probability for the set of segment endpoints $\mathbf{X}$, lying on the plane $(\mathbf{n}, \mathbf{p})$, is simply the product of individual conditional probabilities:

$$p(\mathbf{X} \mid \mathbf{n}, \mathbf{p}) = \prod_i p(\mathbf{x}_i \mid \mathbf{n}, \mathbf{p}) = \prod_i N(\mathbf{n}^T(\mathbf{x}_i - \mathbf{p}), \Lambda_i) \tag{5.5}$$

Our first task is to estimate a point $\mathbf{p}$ on the plane. Since we have no prior knowledge, estimation of the point will be based on a Maximum Likelihood (ML) approach.

$$\hat{\mathbf{p}}_{ML} = \arg\max_{\mathbf{p}} p(\mathbf{X} \mid \mathbf{n}, \mathbf{p}) \tag{5.6}$$

Maximization of $p(\mathbf{X} \mid \mathbf{n}, \mathbf{p})$ is equivalent to maximization of $\ln\left[p(\mathbf{X} \mid \mathbf{n}, \mathbf{p})\right]$, since taking the logarithm does not change the result. Thus, we can estimate $\hat{\mathbf{p}}_{ML}$ as

$$\begin{aligned}
\hat{\mathbf{p}}_{ML} &= \arg\max_{\mathbf{p}} \; \ln\left[p(\mathbf{X} \mid \mathbf{n}, \mathbf{p})\right] && (5.7)\\
&= \arg\max_{\mathbf{p}} \; \ln\left[\prod_i N(\mathbf{n}^T(\mathbf{x}_i - \mathbf{p}), \Lambda_i)\right] && (5.8)\\
&= \arg\max_{\mathbf{p}} \; \sum_i \left[-\frac{1}{2}\,\mathbf{n}^T(\mathbf{x}_i - \mathbf{p})\Lambda_i^{-1}(\mathbf{x}_i - \mathbf{p})^T\mathbf{n}\right] && (5.9)\\
&= \arg\min_{\mathbf{p}} \; \sum_i \left[\frac{1}{2}\,\mathbf{n}^T(\mathbf{x}_i - \mathbf{p})\Lambda_i^{-1}(\mathbf{x}_i - \mathbf{p})^T\mathbf{n}\right] && (5.10)
\end{aligned}$$

Next, to ensure that we get a non-zero solution, the surface normal is constrained to be a normalized vector. A normalization term $(1 - \mathbf{n}^T\mathbf{n})$ scaled by the Lagrange multiplier λ is added to Equation 5.10:

$$\hat{\mathbf{p}}_{ML} = \arg\min_{\mathbf{p}} \left\{\frac{1}{2}\,\mathbf{n}^T\sum_i\left[(\mathbf{x}_i - \mathbf{p})\Lambda_i^{-1}(\mathbf{x}_i - \mathbf{p})^T\right]\mathbf{n} + \lambda(1 - \mathbf{n}^T\mathbf{n})\right\} \tag{5.11}$$

Let us define $D_S(\mathbf{n}, \mathbf{p})$ as the term to be optimized

$$D_S(\mathbf{n}, \mathbf{p}) = \frac{1}{2}\,\mathbf{n}^T\sum_i\left[(\mathbf{x}_i - \mathbf{p})\Lambda_i^{-1}(\mathbf{x}_i - \mathbf{p})^T\right]\mathbf{n} + \lambda(1 - \mathbf{n}^T\mathbf{n}) \tag{5.12}$$

Computing the gradient of $D_S(\mathbf{n},\mathbf{p})$ with respect to $\mathbf{p}$, we obtain

$$\nabla_{\mathbf{p}}\, D_S(\mathbf{n},\mathbf{p}) = \mathbf{n}^T \left[\sum_i \left(\mathbf{p}\Lambda_i^{-1} - \mathbf{x}_i\Lambda_i^{-1}\right)\right]\mathbf{n} \tag{5.13}$$

Setting the gradient to zero leads to

$$\mathbf{p}\sum_i \Lambda_i^{-1} = \sum_i\ \mathbf{x}_i\Lambda_i^{-1} \tag{5.14}$$

The desired point $\mathbf{p}$ can therefore be estimated as

$$\hat{\mathbf{p}}_{ML} = \frac{\sum_i\ \mathbf{x}_i\Lambda_i^{-1}}{\sum_i \Lambda_i^{-1}} \tag{5.15}$$

If all the covariances are identical, i.e. $\Lambda_i^{-1} = c,\ \forall i$, the solution reduces to

$$\hat{\mathbf{p}}_{ML} = \frac{\sum_i\ \mathbf{x}_i}{N} \tag{5.16}$$

where N in the denominator is the number of contributing segment endpoints for the current surface. In other words, surface point $\mathbf{p}$ can be estimated simply as the average of the bounding endpoints $\mathbf{x}_i$.

Next, we need to estimate the surface normal $\mathbf{n}$, using the estimate of $\hat{\mathbf{p}}$ computed above. As before, the estimation of normal $\mathbf{n}$ will be based on a Maximum Likelihood (ML) approach.

$$\hat{\mathbf{n}}_{ML} = \arg\max_{\mathbf{n}}\, p(\mathbf{X} \mid \mathbf{n},\mathbf{p}) \tag{5.17}$$

$$= \arg\min_{\mathbf{n}}\, D_S(\mathbf{n},\mathbf{p}) \tag{5.18}$$

Computing the gradient of $D_S(\mathbf{n},\mathbf{p})$ with respect to $\mathbf{n}$, we get

$$\nabla_{\mathbf{n}}\, D_S(\mathbf{n},\mathbf{p}) = \mathbf{n}^T \left[\sum_i \left(\mathbf{x}_i - \mathbf{p}\right)\Lambda_i^{-1}\left(\mathbf{x}_i - \mathbf{p}\right)^T\right] - \lambda\mathbf{n}^T \tag{5.19}$$

where the Lagrange multiplier λ constrains the vector $\mathbf{n}$ to be unit in length.

Setting the gradient to zero leads to

$$\left[\sum_i (\mathbf{x}_i - \mathbf{p})\Lambda_i^{-1}(\mathbf{x}_i - \mathbf{p})^T\right]\mathbf{n} = \lambda\mathbf{n} \tag{5.20}$$

This equation is a standard eigenvalue problem. To solve this minimization problem, one needs to find the eigenvector $\mathbf{n}$ with the smallest eigenvalue for the left-hand side matrix. The solution $\hat{\mathbf{n}}_{ML}$ can be determined by the method of SVD.

The residual of the plane fitting process is computed by substituting the estimates of $\mathbf{n}$ and $\mathbf{p}$ into Equation 5.12.

5.4 Visibility Constraint Tests

As we have seen in Chapter 2, feature extrusions can often intersect in 3D space by accident. Many of these accidental intersections occur in violation of a simple visibility constraint – that feature extrusions should not penetrate surfaces to form correspondence hypotheses [10]. If we can detect these violations early, many invalid hypotheses may be quickly eliminated, which will greatly reduce the computational burden of the system.

In this section, we describe how violation of the visibility constraint can be detected. An important assumption we make is that there is no transparent surface of interest in the scene. While this assumption may seem too strong since glass windows are commonplace in urban environment, our observations suggest that, in most cases, the glass windows appear to be more opaque than transparent. Besides glass windows, there are not too many other large and flat transparent surfaces.

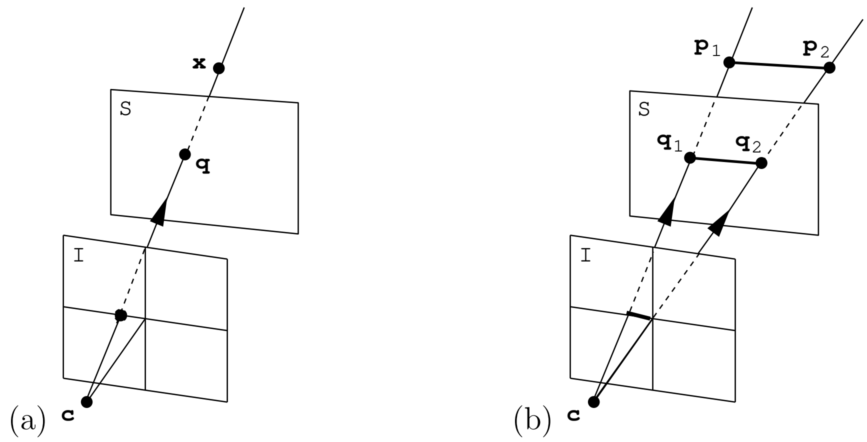


Figure 5.5: Visibility constraint tests: (a) vertex intersection test, (b) segment intersection test.

To test for violation by a vertex hypothesis, we begin by examining its supporting vertex extrusions. For each vertex extrusion, we compute a point $\mathbf{q}$ on the planar surface S, intersected by the extrusion ray, not knowing whether the intersection occurs in front, or behind the vertex hypothesis. See Figure 5.5(a). Next we compute d, the distance from camera center $\mathbf{c}$ to surface point $\mathbf{q}$, minus the distance from camera center $\mathbf{c}$ to the center of vertex hypothesis $\mathbf{x}$:

$$d = \|\mathbf{q} - \mathbf{c}\| - \|\mathbf{x} - \mathbf{c}\| \tag{5.21}$$

If $d > 0$, the point $\mathbf{x}$ is closer to the camera than $\mathbf{q}$, so it is clear that there is no violation of the surface visibility constraint. If $-d > d_v$, where d_v is a predetermined threshold, we conclude that there is a violation of the surface visibility constraint. Anything in between is inconclusive.

To test for violation by a segment hypothesis, we begin by examining its supporting segment extrusions. For each segment extrusion, we first compute the line segment $(\mathbf{q}_1, \mathbf{q}_2)$ on the planar surface S, intersected by the extrusion wedge, not knowing whether the intersection occurs in front or behind the segment hypothesis. See Figure 5.5(b). Next we compute d, the distance from camera center $\mathbf{c}$ to surface segment $(\mathbf{q}_1, \mathbf{q}_2)$, minus the distance from camera center $\mathbf{c}$ to the segment hypothesis $(\mathbf{p}_1, \mathbf{p}_2)$:

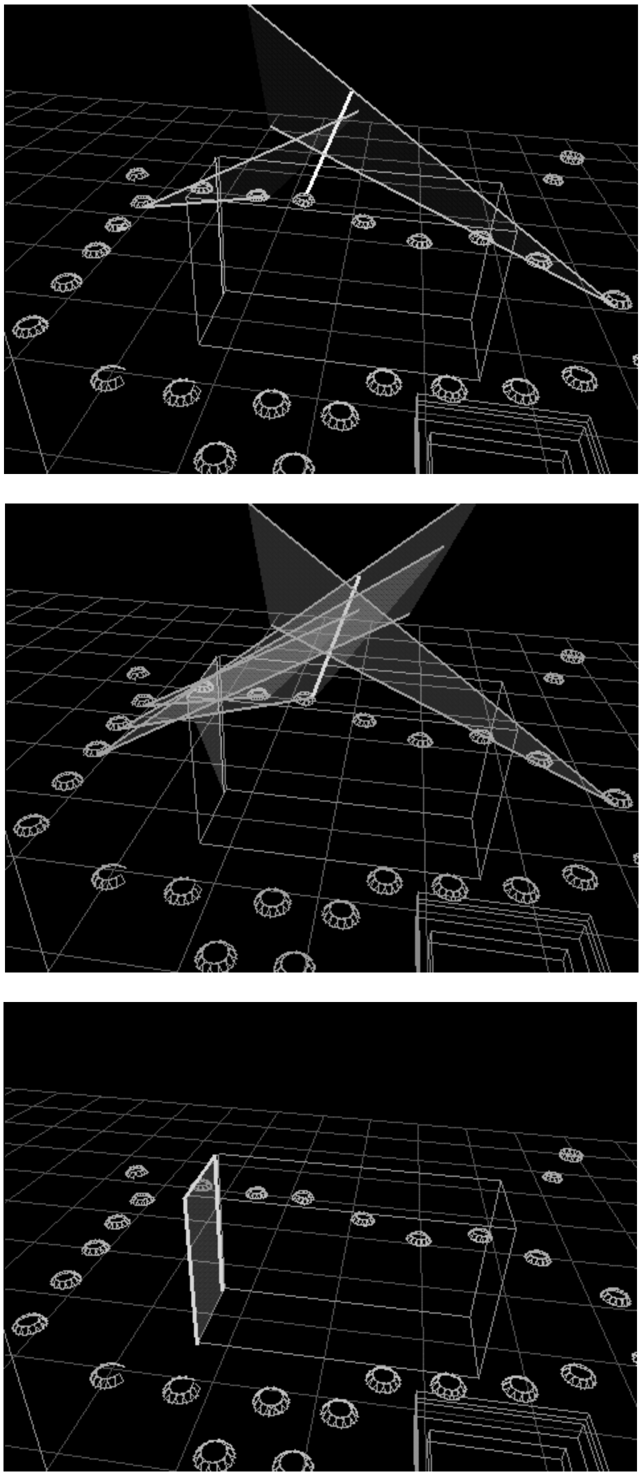
$$d = \|(\mathbf{q}_1, \mathbf{q}_2) - \mathbf{c}\| - \|(\mathbf{p}_1, \mathbf{p}_2) - \mathbf{c}\| \tag{5.22}$$

If $d > 0$, the segment $(\mathbf{p}_1, \mathbf{p}_2)$ is closer to the camera than $\mathbf{q}$, so it is clear that there is no violation of the surface visibility constraint. If $-d > d_v$, where d_v is a pre-determined threshold, we conclude that there is a violation of the surface visibility constraint. Like before, anything in between is inconclusive.

A visualization of the enforcement of surface visibility constraint is presented in Figure 5.6. In (a), we observe the formation of an invalid segment hypothesis. At this point, no surface has formed yet. In (b), the invalid segment hypothesis recruits additional segment extrusions consistent with it. This happens even though a blocking surface hypothesis has been formed on the left side of the building. Until the surface hypothesis is confirmed, no surface visibility constraint will be enforced. In (c), as the blocking surface becomes confirmed, violations of the visibility constraint are promptly detected. The result is the demise of the invalid segment hypothesis.

5.5 Summary

In this chapter, we have accomplished the following tasks:



(a)

(b)

(c)

Figure 5.6: Enforcement of surface visibility constraint. (a) Formation of a segment hypothesis. (b) The hypothesis grows despite the formation of a blocking surface hypothesis. (c) The segment hypothesis is killed when the surface hypothesis is confirmed.

1. We observed that in our multi-level system, information processing no longer happens purely on a local level. Surface level information can provide important constraints on low-level feature correspondence.

2. We outlined the major steps involved in surface computation through a flow diagram. Two major tasks are involved – surface hypothesis processing, and the enforcement of surface visibility constraints.

3. We derived the solution to planar surface estimation using probabilistic modeling and Maximum Likelihood estimation. The solution involves two separate steps: 1) estimating a point on the surface, and 2) estimating the surface normal.

4. We described two tests for checking violation of surface visibility constraint. One checks for extrusions associated with vertex hypotheses, and the other checks for extrusions associated with segment hypotheses.

Chapter 6

Reconstruction Results

The main objective of our reconstruction system is to perform large-scale 3D reconstruction of the urban environment, and generate output suitable for simulation and visualization. Large building structures are the main targets of interest for us.

In this section, we first describe the graphical user interface of our reconstruction system. This is helpful for understanding how the results are obtained. We then present incremental reconstruction results of our system operating on two large data sets of the Technology Square, one consisting of synthetic images, and the other real photographs. The synthetic image data set is smaller, and easier in the sense that the image and pose data are relatively accurate, and not corrupted by noise and measurement errors. In contrast, the real image data set is more challenging because of its bigger size, variable lighting conditions, and noisy image and pose measurements.

Our results indicate that the system works reasonably well under these conditions, but is dependent on the coverage of input images and the quality of feature detection. Analysis also shows that the visibility constraints of confirmed surfaces play a significant role in improving the correspondence process.

6.1 Graphical User Interface

The graphical user interface (GUI) of our reconstruction system is designed to facilitate the visualization and control of the reconstruction process. In this section, we provide a brief overview of the interface and the possible user controls.

6.1.1 Main Image Window

The GUI has two main windows – the main image window, and the virtual world window. In the main image window, a chosen input image is displayed (see Figure 6.1). The user can select a specific image to display using one of the two sliders at the bottom of the bottom. The upper slider loops over a pre-selected set of input images that are computed to be within a specific distance of the reconstruction focus region. The lower slider loops over the entire set of input images.

From the menu labeled "File" on the panel of the main image window, the user can select one of the following four actions:

1. Open Camera Window – open the virtual world window.
2. Read Data for Current Image – read image and feature data from disk for the current image.
3. Close Image Window – close the main image window (leaving virtual world window).
4. Exit – exit the system.

From the menu labeled "Edit" on the panel in the main image window, the user can edit the parameters of the following processes:

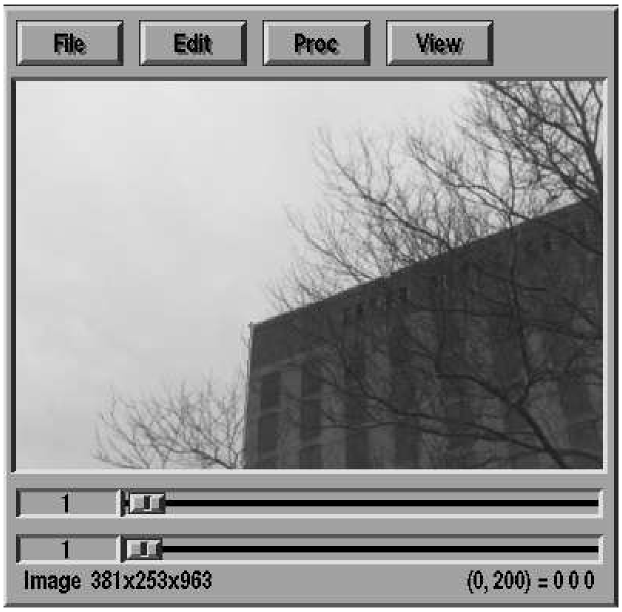


Figure 6.1: The main image window – showing an RGB image

1. Edge Detection Parameters – change the parameters for the Canny edge detection process.

2. Line Detection Parameters – change the parameters for the line extraction process.

3. Vertex Detection Parameters – change the parameters for the vertex extraction process.

4. Reconstruction Parameters – change the parameters for the 3D reconstruction process.

Through the menu labeled "Proc" on the panel of the main image window, the user can initiate and control the following processes:

1. Image Loop – brings up a control panel for automatically looping over the pre-selected set of images.

2. Feature Detection – brings up a control panel for executing edge detection, line extraction, and vertex extraction.

3. 3D Reconstruction – brings up a control panel for executing 3D reconstruction.

The type of image displayed in the main image window can be changed. This is done by selecting one of the following image types from the "View" menu in the panel:

1. RGB Image – original color input image.
2. Edge Image – Canny edge output of the RGB input.
3. Line Image – lines extracted from the RGB input.
4. Line/Vertex Image – lines and vertices extracted from the RGB input.
5. Edge/RGB Image – Canny edge output overlaid on the RGB input.
6. Edge/Line Image – extracted lines overlaid on the Canny edge output.
7. Edge/Line/Vertex Image – extracted lines and vertices overlaid on the Canny edge output.
8. RGB/Line Image – extracted lines overlaid on the RGB input.
9. RGB/Line/Vertex Image – extracted lines and vertices overlaid on the RGB input.

6.1.2 Virtual World Window

The purpose of the virtual world window is to allow the user to visualize and interact with the 3D environment under reconstruction (see Figure 6.2). The input images are positioned in the space precisely according to their corresponding camera pose. Input feature extrusions, current reconstruction results, and pre-specified 3D models can be

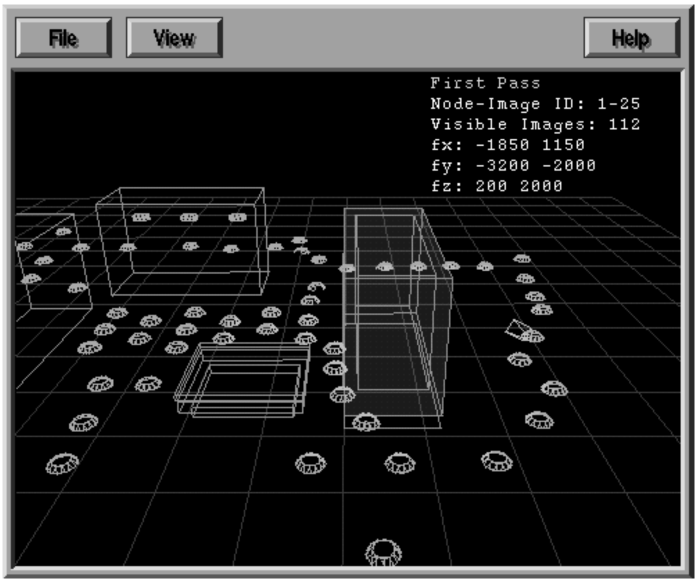


Figure 6.2: The virtual world window – showing input image positions, pre-defined building models, and the reconstruction focus region.

selectively displayed. Using the mouse, the user can freely navigate in the 3D space to see the 3D environment from different angles.

Moreover. the use can point and click on many objects in the scene for more information. If the user picks on a specific input image in the scene, the image ID is displayed in the upper right corner of the virtual view window, and the corresponding image is displayed in the main image window. If the user picks on a specific feature extrusion in the scene, the extrusion ID is displayed in the upper right corner of the virtual view window, all linked correspondence hypotheses and elements become highlighted, and the corresponding image feature is displayed in the main image window. If the user picks on a hypothesis or element, the object information is displayed in the upper left corner, and all supporting extrusions are immediately highlighted.

From the “View” menu on the panel of the virtual world window, the user can change various parameters controlling object magnification, highlighting, and the reconstruction focus region. The reconstruction focus region is displayed as a rectangular volume in the virtual world window. To perform 3D reconstruction in a spatially focused manner, only input images that intersect with the volume of the focus region are used in the reconstruction process.

The “Help” menu on the panel brings up a help panel listing all the key mappings that control the motion of viewing camera and the visibility of several pre-defined objects in the virtual world window.

6.1.3 Reconstruction Control Panel

The reconstruction control panel controls the execution of the 3D reconstruction process. A snapshot of the reconstruction control panel is displayed in Figure 6.3. The upper part of the panel consists of ten labeled buttons that when pressed down would turn on the visibility of a specific class of reconstructed objects. To the right of each button is a field displaying the current number of active objects in each class. The fields are constantly updated during the reconstruction process.

The middle part of the panel presents a number of switches that controls several aspects of the reconstruction process, such as the input images to use, the processing speed, output clipping by focus region, and surface visibility blocking. They can be adjusted during the reconstruction process.

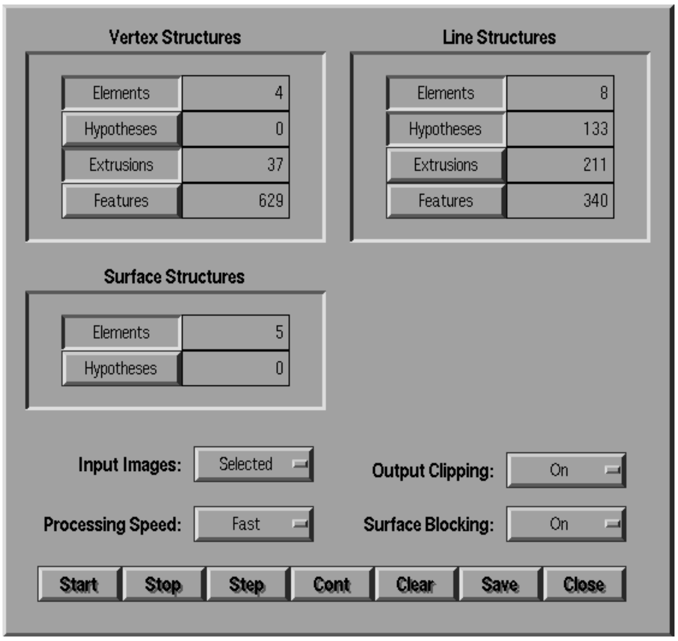


Figure 6.3: Reconstruction Control Panel – the buttons indicate 4 vertex elements, 37 vertex extrusions, 8 segment elements, 133 segment hypotheses, and 5 surface elements are visible in the virtual world window.

At the bottom part of the panel are seven buttons that controls the execution of reconstruction process. The buttons are

1. Start button – starts the reconstruction process from the beginning.
2. Stop button – interrupts and stops the current reconstruction process.
3. Step button – steps through one cycle of the reconstruction process and then stops.
4. Cont button – continues the currently stopped reconstruction process.
5. Clear button – clears the current state of the reconstruction process.
6. Save button – saves the current state of the reconstruction process into a file.
7. Close button – closes the reconstruction control panel.

6.2 Synthetic Image Test

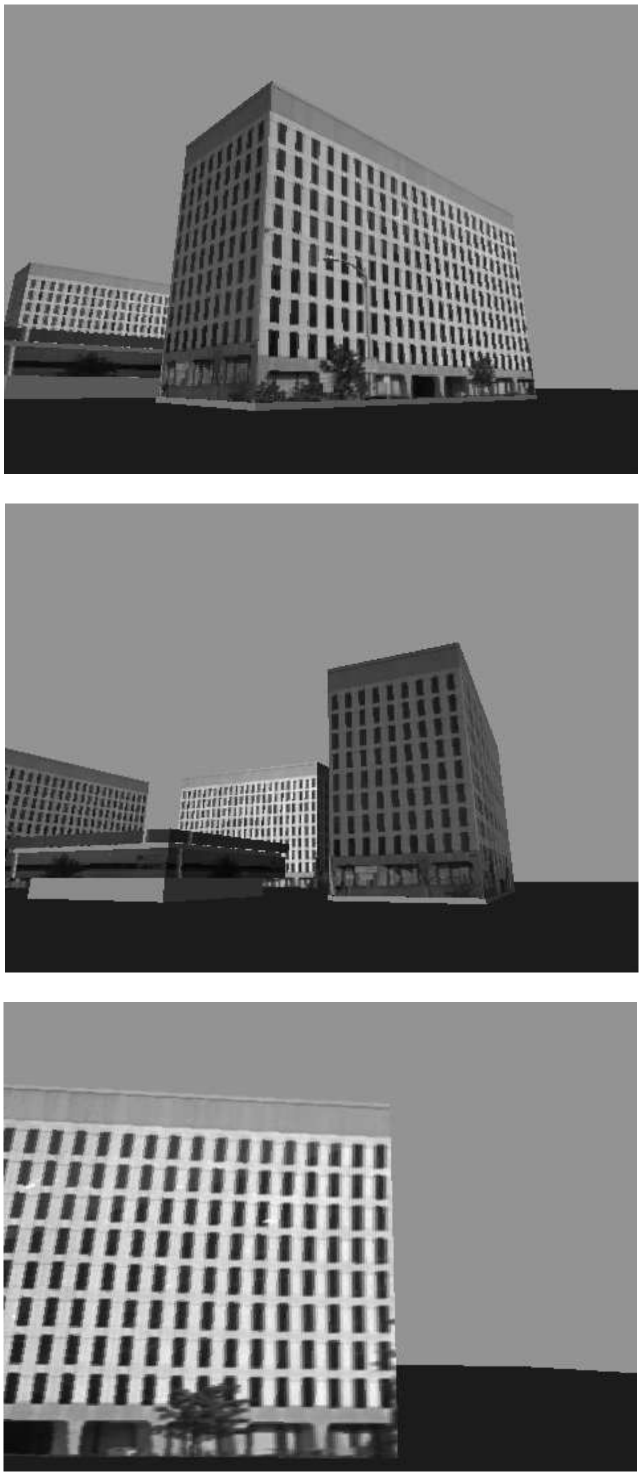
We tested our system on 100 480x360 color images generated from building models of the Technology Square. The 3D wireframe models of the buildings were obtained from the blueprints and then manually constructed. The image data set was synthesized using computer graphics techniques with realistic lighting conditions and texture mapping. The camera positions of the synthetic images were computed based on a circular path around the center of the Square.

Three images from the synthetic image data set are shown in Figure 6.4. In the foreground of Figure 6.4(a) is building 545, with building 575 behind it to the left. In Figure 6.4(b), building 545 is the tall building closest to the camera, with buildings 565 and 575 in the background. A close-up image of building 575 is shown in Figure 6.4(c).

The feature detection process of an image is visualized in Figure 6.5. In Figure 6.5(b), the gradient magnitude of the image is shown. In Figure 6.5(c), the output of Canny edge detection is shown. In Figures 6.5(d),(e),(f), line segments extracted with minimum length thresholds of 60, 80, and 100 pixels are displayed.

The reconstruction setup for the synthetic image data set is shown in Figure 6.6. In Figure 6.6(a), the positions of the images are visualized along with the building models. The vertex extrusions and segment extrusions used in the reconstruction process are visualized in Figures 6.6(b) and (c), respectively.

An overview of the reconstruction process is visualized in Figure 6.7. Building 545 is analyzed and recovered first, followed by building 565, and then building 575.



(a)

(b)

(c)

Figure 6.4: Three images from the synthetic image data set. (a) an image of building 545 with building 575 behind it. (b) an image of building 545 in the front, and buildings 565 and 575 in the background to the left. (c) a close-up image of building 575.

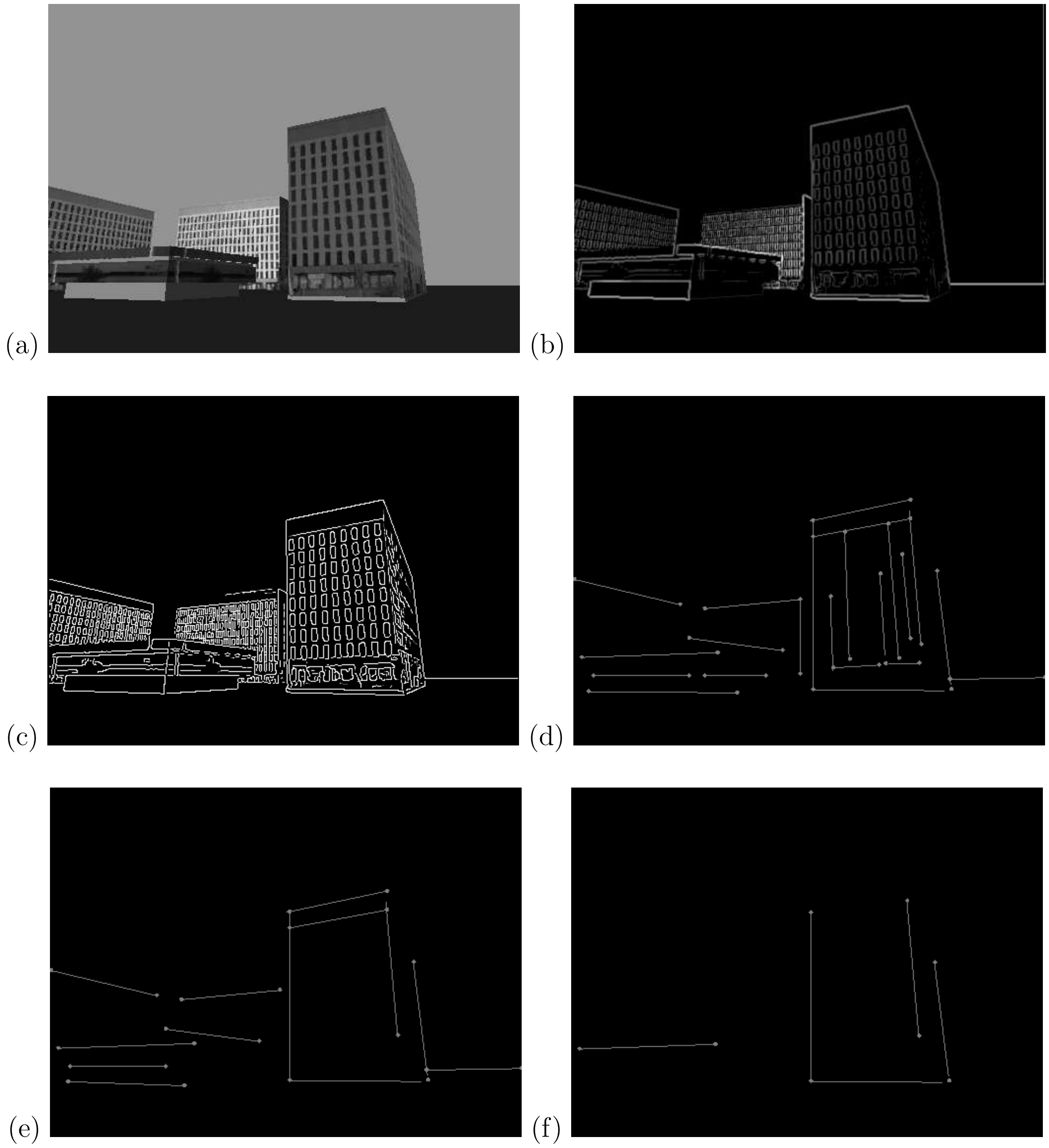
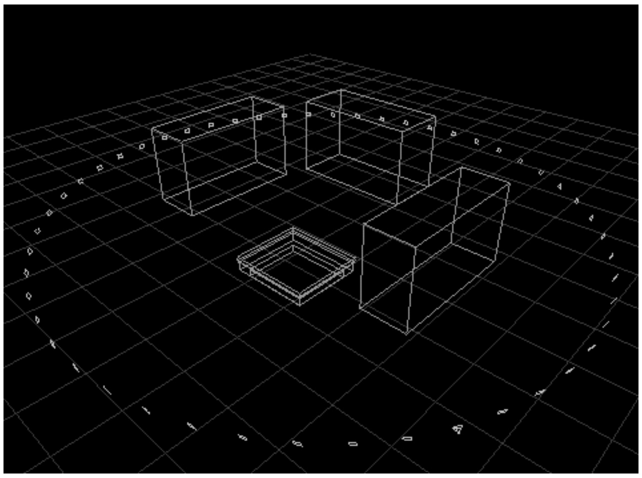
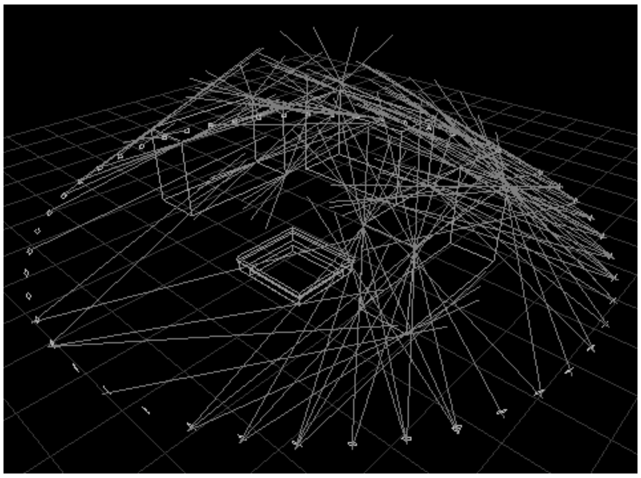


Figure 6.5: Line segments detected for a synthetic image with different minimum segment length threshold. (a) The original image. (b) The gradient magnitude image. (c) Canny edge output. (d) Length threshold of 60 pixels, (e) 80 pixels, (f) 100 pixels.

(a)



(b)



(c)

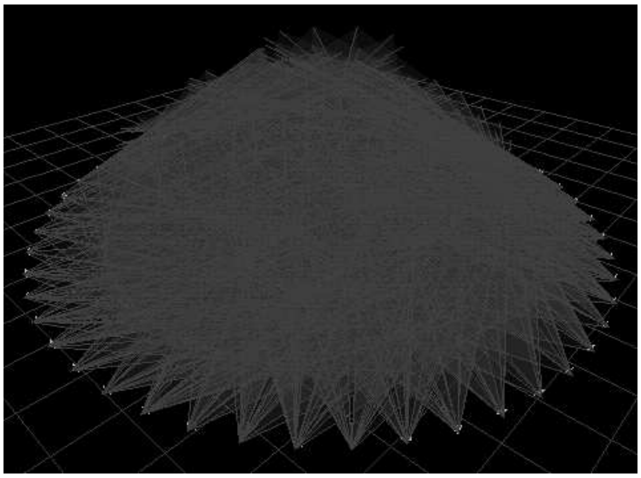
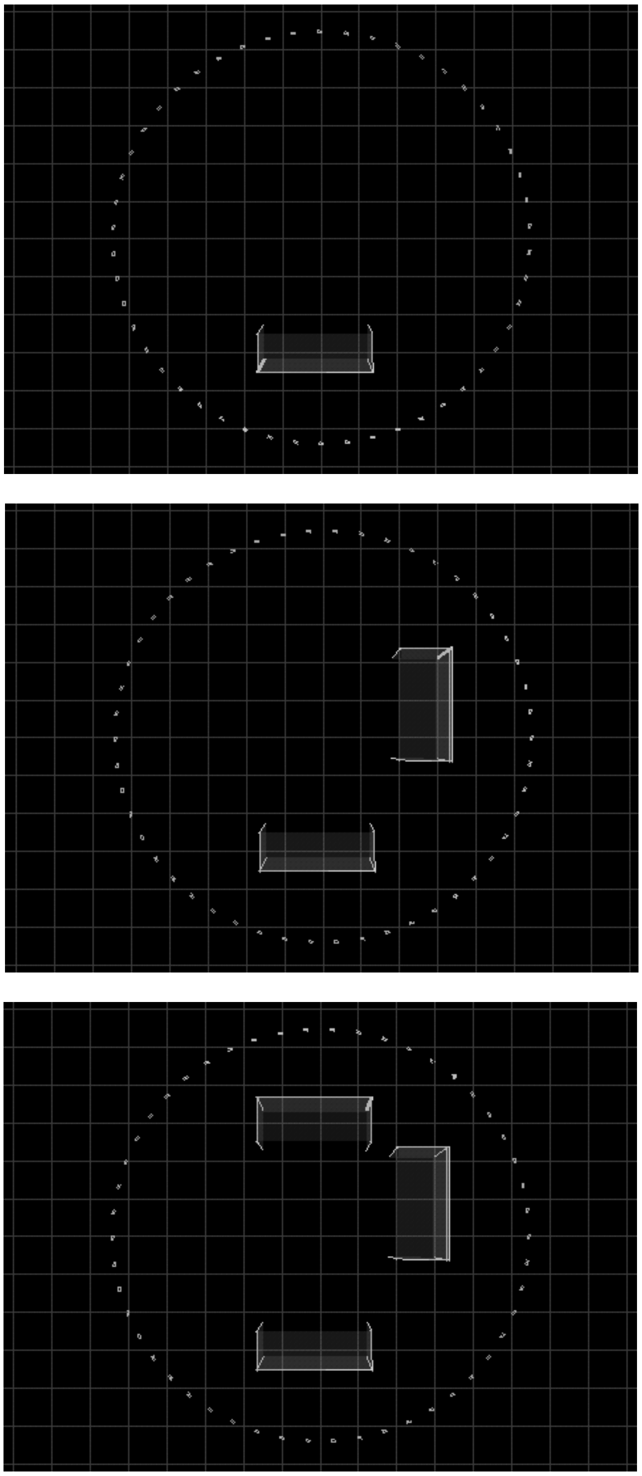


Figure 6.6: Synthetic image reconstruction setup. (a) Image positions and building models. (b) Input vertex extrusions. (c) Input segment extrusions.



(a)

(b)

(c)

Figure 6.7: An overview of the reconstruction process. (a) Building 545 recovered. (b) Building 565 recovered. (c) Building 575 recovered.

The results presented in this section were obtained using an SGI O2 Visual Workstation running on a MIPS R10000 processor with 256MB of main memory. The average run-time for reconstructing each building is about 15 seconds.

6.2.1 Building 545 Reconstruction

Incremental reconstruction of building 545 is visualized in Figure 6.8. Figures 6.8(a),(c), and (e) show surface hypotheses, surface elements, and segment elements in different stages of reconstruction. Figures 6.8(b),(d), and (f) have segment hypotheses added to them.

The input/output relationship of the reconstruction process is displayed in Figure 6.9. The vertex extrusions and segment extrusions used as input to the system are shown in Figures 6.9(a) and (b) respectively. The output surface hypotheses, surface elements, and segment elements are shown in Figure 6.9(c).

The reconstructed building 545 is incomplete, missing a line segment between the roof and the inner side surface. This is due to a lack of sufficient images observing it. The connections between several recovered segment elements and their corresponding extrusions are visualized in Figure 6.10.

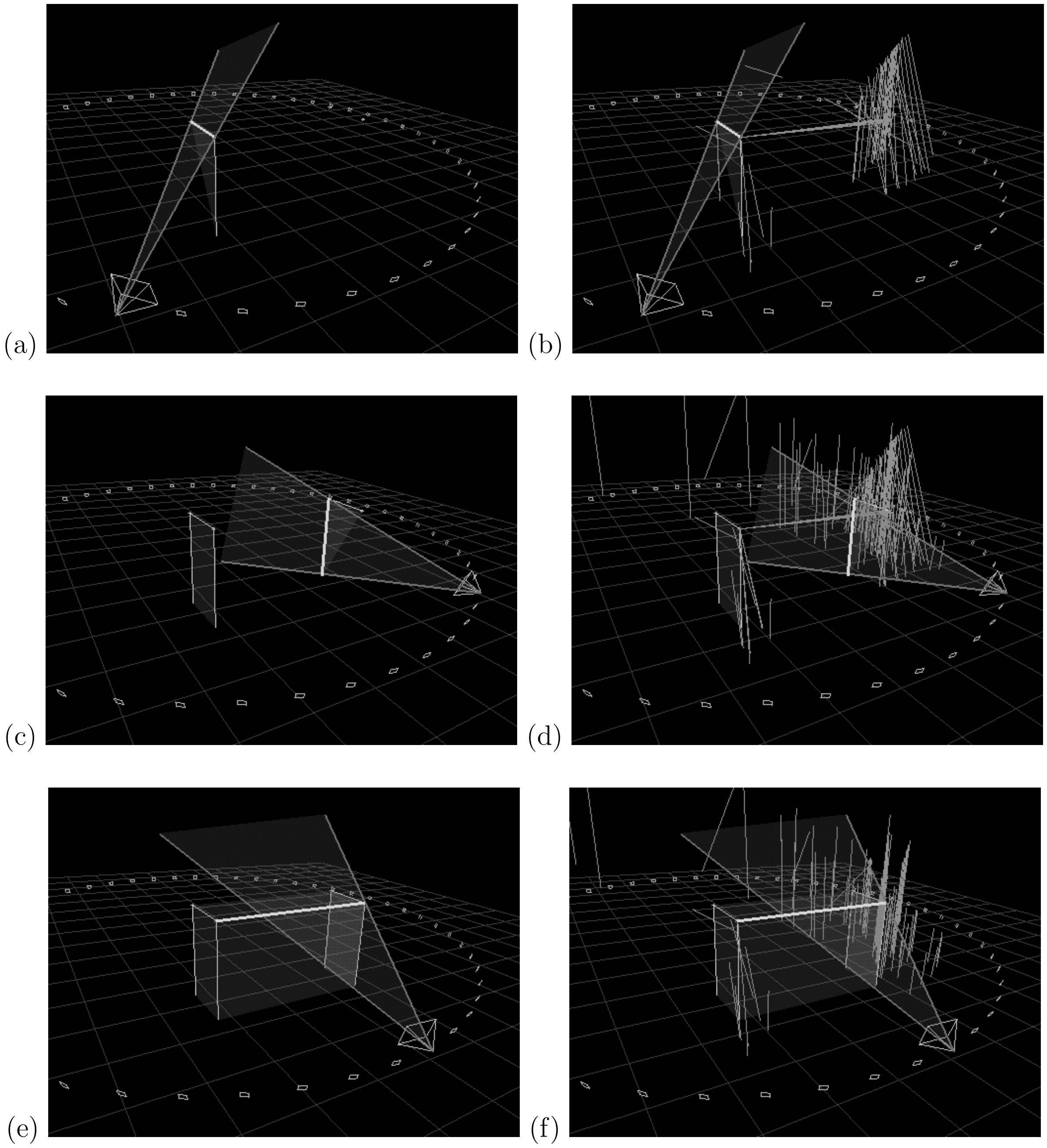
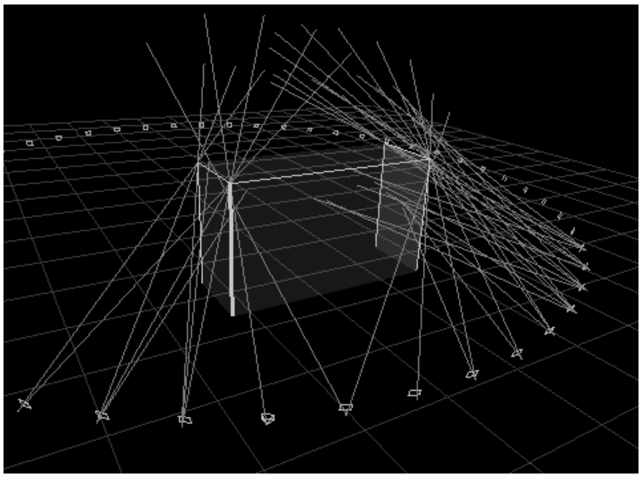
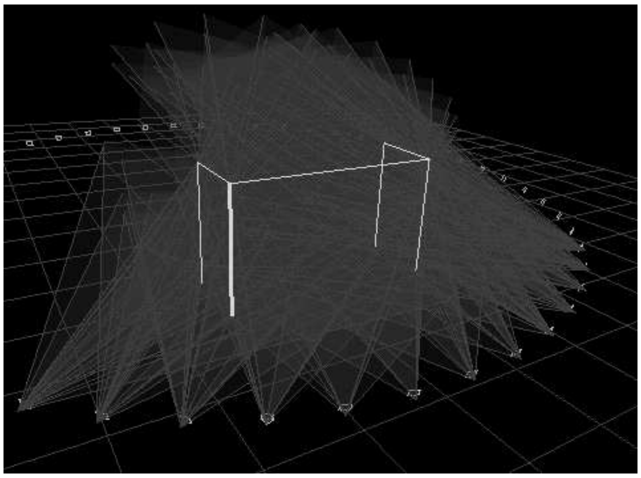


Figure 6.8: Reconstruction of building 545 in progress. (a),(c),(e) Surface hypotheses, surface elements and segment elements. (b),(d),(f) Segment hypotheses added to above.

(a) 

(b) 

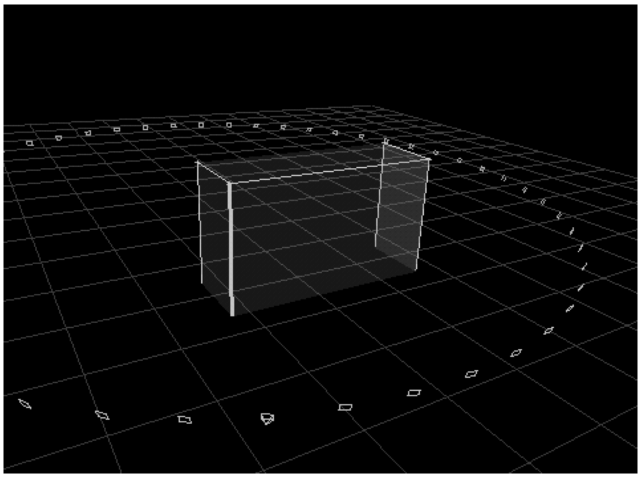
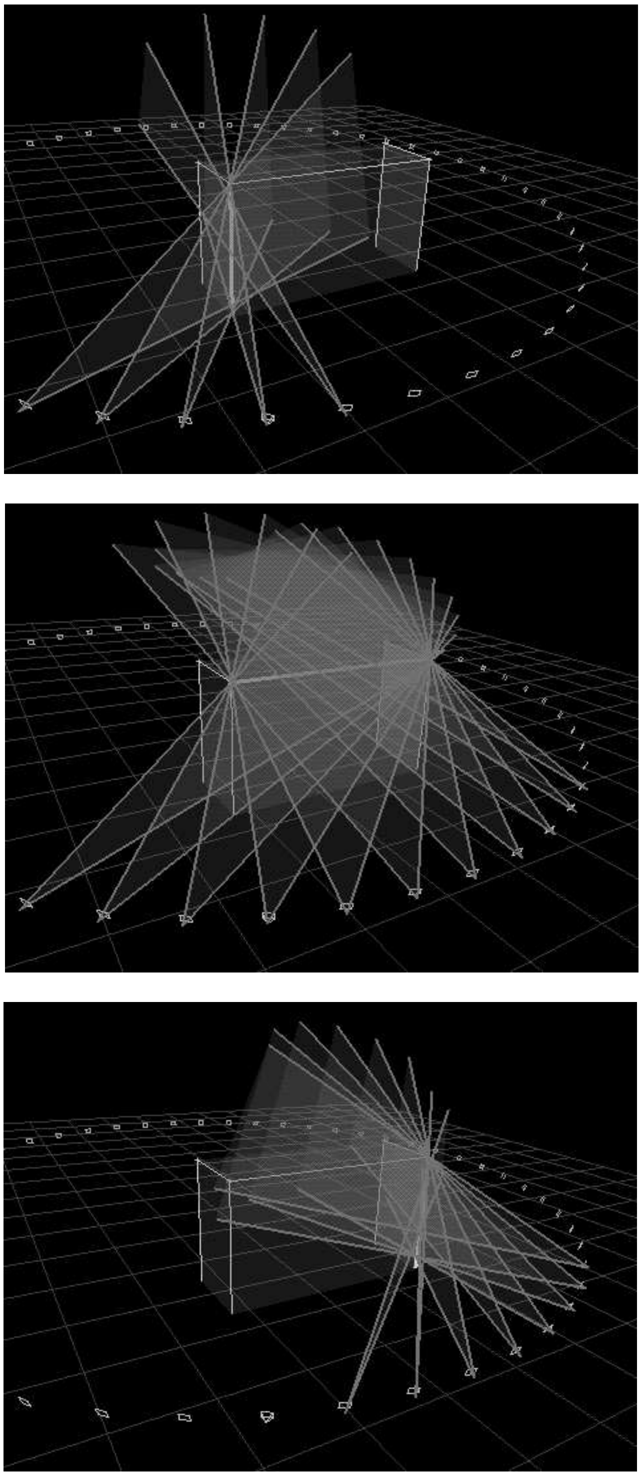
(c) 

Figure 6.9: Input and output for building 545. (a) Input vertex extrusions. (b) Input segment extrusions. (c) Output surface hypotheses, surface elements, and segment elements.



(a) (b) (c)

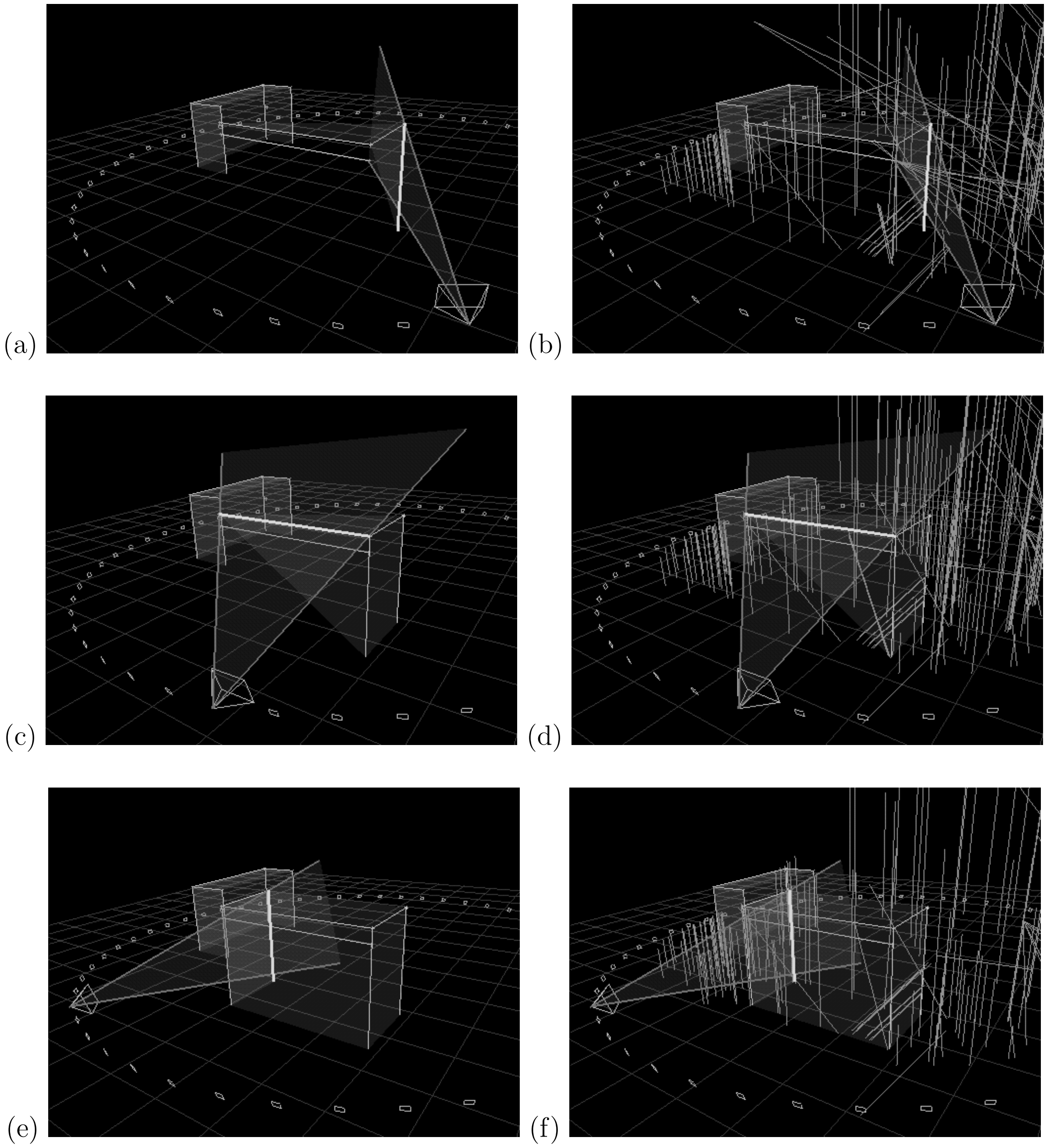
Figure 6.10: Focus on recovered segments of building 545. (a) A vertical segment. (b) A horizontal segment. (c) A vertical segment.

6.2.2 Building 565 Reconstruction

Incremental reconstruction of building 565 is visualized in Figure 6.11. Figures 6.11(a),(c), and (e) show surface hypotheses, surface elements, and segment elements in different stages of reconstruction. Figures 6.11(b),(d), and (f) have segment hypotheses added to them.

The input/output relationship of the reconstruction process is displayed in Figure 6.12. The vertex extrusions and segment extrusions used as input to the system are shown in Figures 6.12(a) and (b) respectively. The output surface hypotheses, surface elements, and segment elements are shown in Figure 6.12(c).

The reconstructed building 565 is incomplete, also missing a line segment between the roof and the inside surface. This is due to a lack of sufficient images observing it. It is interesting to note that the system has recovered an additional line segment on the outer side surface. The connections between several recovered segment elements and their corresponding extrusions are visualized in Figure 6.13.



(a) (b) (c) (d) (e) (f)

Figure 6.11: Reconstruction of building 565 in progress. (a),(c),(e) Surface hypotheses, surface elements and segment elements. (b),(d),(f) Segment hypotheses added to above.

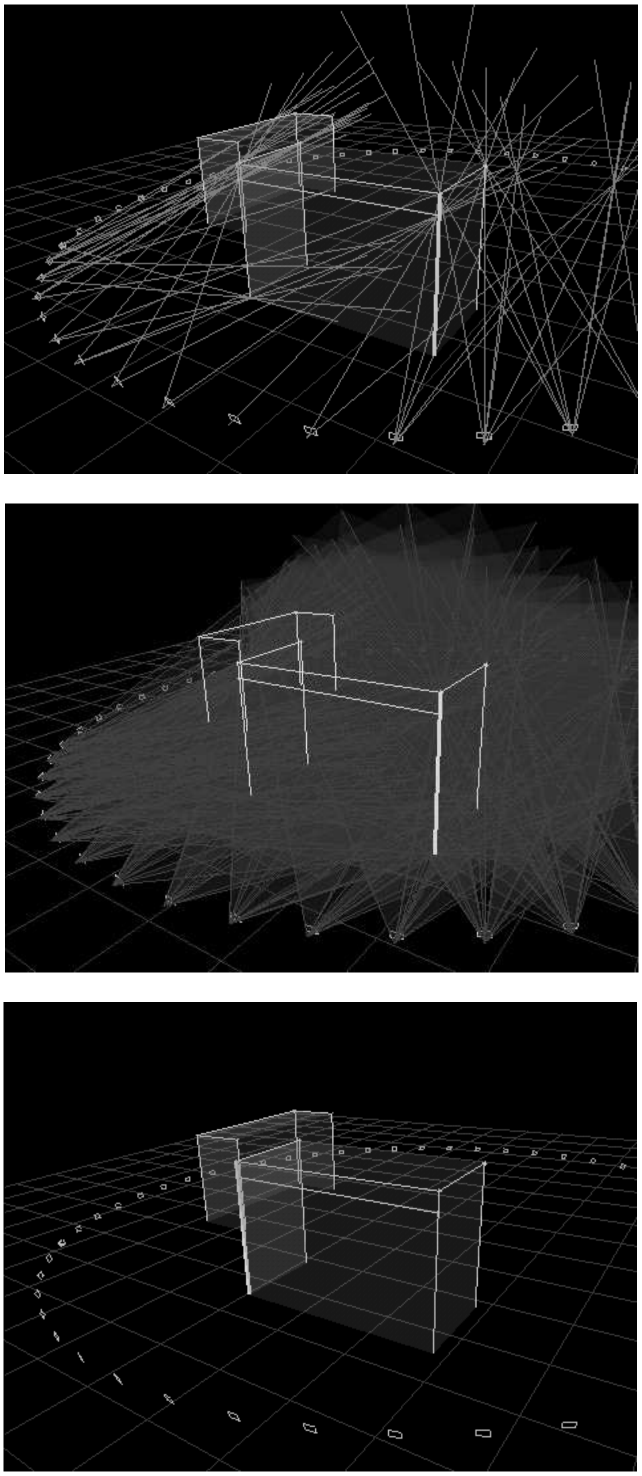
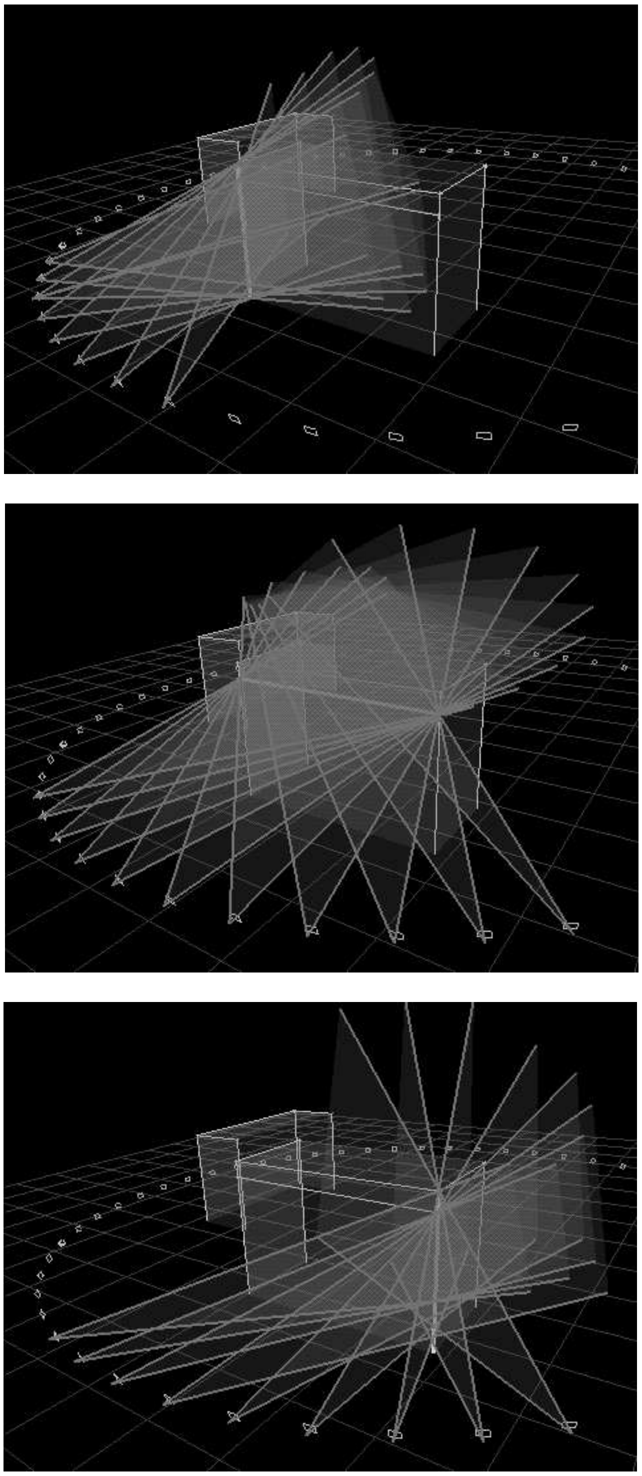

(a)
(b)
(c)

Figure 6.12: Input and output for building 565. (a) Input vertex extrusions. (b) Input segment extrusions. (c) Output surface hypotheses, surface elements, and segment elements.



(a) (b) (c)

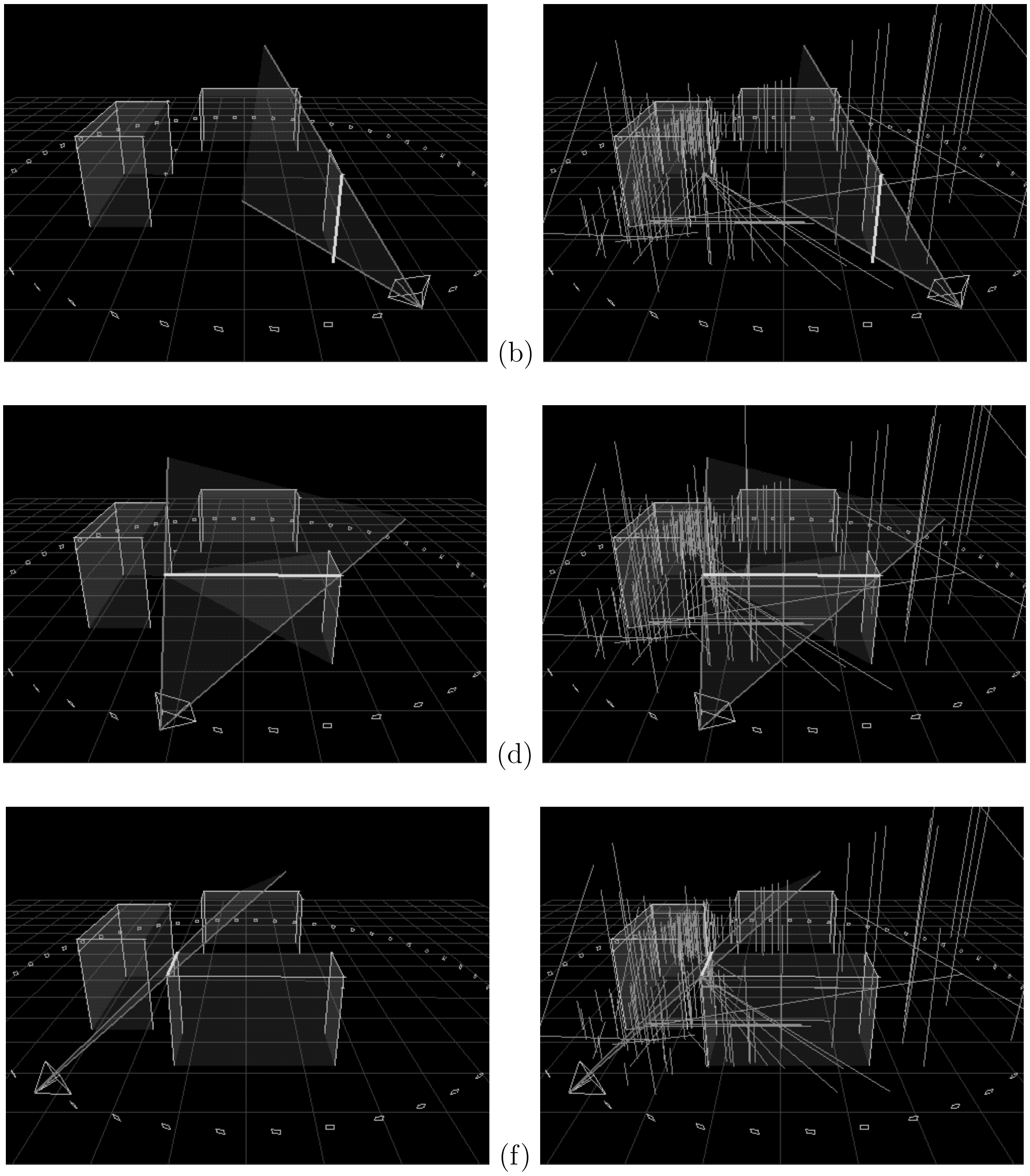
Figure 6.13: Focus on recovered segments of building 565. (a) A vertical segment. (b) A horizontal segment. (c) A vertical segment.

6.2.3 Building 575 Reconstruction

Incremental reconstruction of building 575 is visualized in Figure 6.14. Figures 6.14(a),(c), and (e) show surface hypotheses, surface elements, and segment elements in different stages of reconstruction. Figures 6.14(b),(d), and (f) have segment hypotheses added to them.

The input/output relationship of the reconstruction process is displayed in Figure 6.15. The vertex extrusions and segment extrusions used as input to the system are shown in Figures 6.15(a) and (b) respectively. The output surface hypotheses, surface elements, and segment elements are shown in Figure 6.15(c).

The reconstructed building 575 is incomplete, also missing a line segment between the roof and the inside surface. This again is due to a lack of sufficient images observing it. The connections between several recovered segment elements and their corresponding extrusions are visualized in Figure 6.16.



(a) (b) (c) (d) (e) (f)

Figure 6.14: Reconstruction of building 575 in progress. (a),(c),(e) Surface hypotheses, surface elements and segment elements. (b),(d),(f) Segment hypotheses added to above.

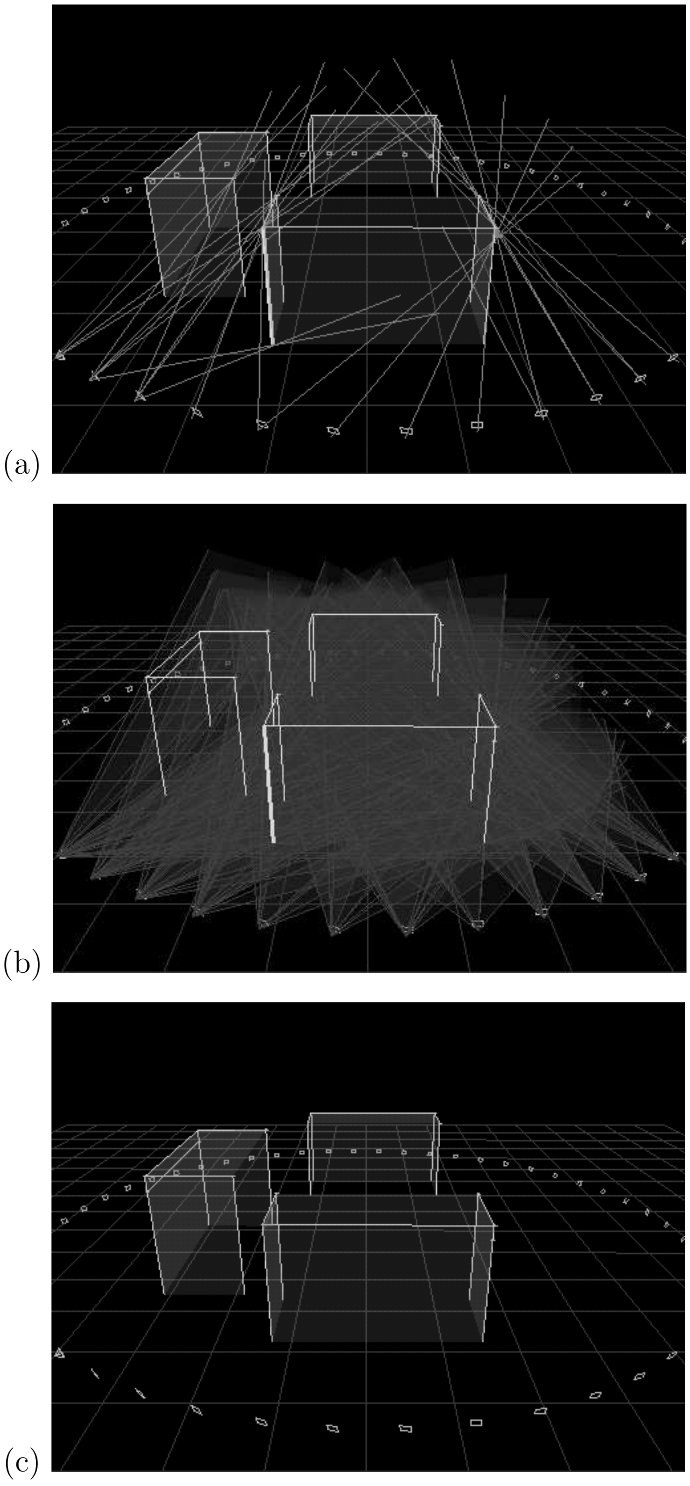


Figure 6.15: Input and output for building 575. (a) Input vertex extrusions. (b) Input segment extrusions. (c) Output surface hypotheses, surface elements, and segment elements.

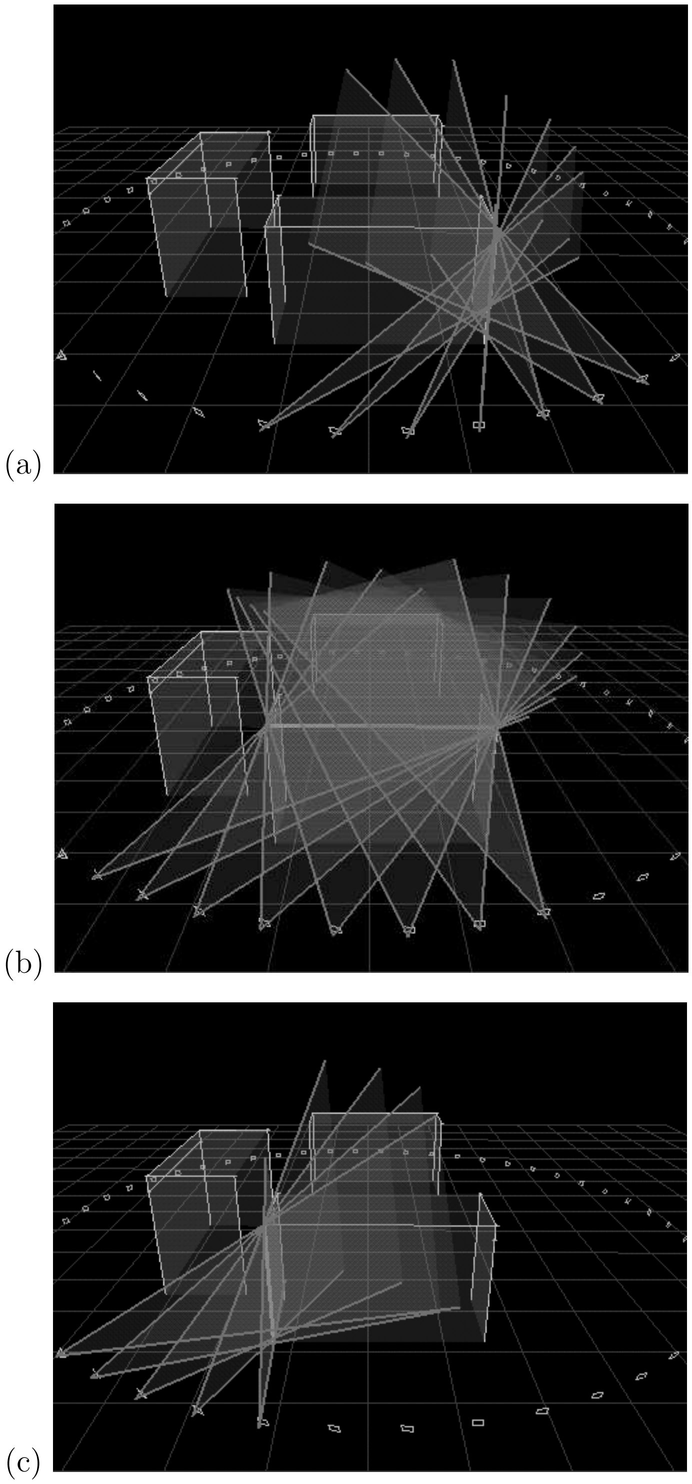


Figure 6.16: Focus on recovered segments of building 575. (a) A vertical segment. (b) A horizontal segment. (c) A vertical segment.

6.3 Real Image Test

Next, we tested our system on 963 381x253 color photographs taken from over 81 positions in and around the Technology Square. The image acquisition process took place over several days, and as a result, the lighting conditions can change dramatically from one photo to the next. Pose information were obtained by manual surveying and instrument measurements, and then refined through a semi-automatic process. Details on the collection process and pose refinement can be found in [13].

Three photographs from the real image data set are shown in Figure 6.17. In Figure 6.17(a), building 545 is visible. In Figure 6.17(b), building 565 is visible. In Figure 6.4(c), building 575 is in the foreground to the left, while building 565 is in the background to the right.

The feature detection process of two images are visualized in Figures 6.18, 6.19. In Figure 6.18,6.19(b), the gradient magnitude of the image is shown. In Figure 6.18,6.19(c), the output of Canny edge detection is shown. In Figures 6.18,6.19(d),(e),(f), line segments extracted with minimum length thresholds of 60, 80, and 100 pixels are displayed.

The reconstruction setup for the real image data set is shown in Figure 6.20. In Figure 6.20(a), the positions of the images are visualized along with the building models. The vertex extrusions and segment extrusions used in the reconstruction process are visualized in Figures 6.20(b) and (c), respectively.

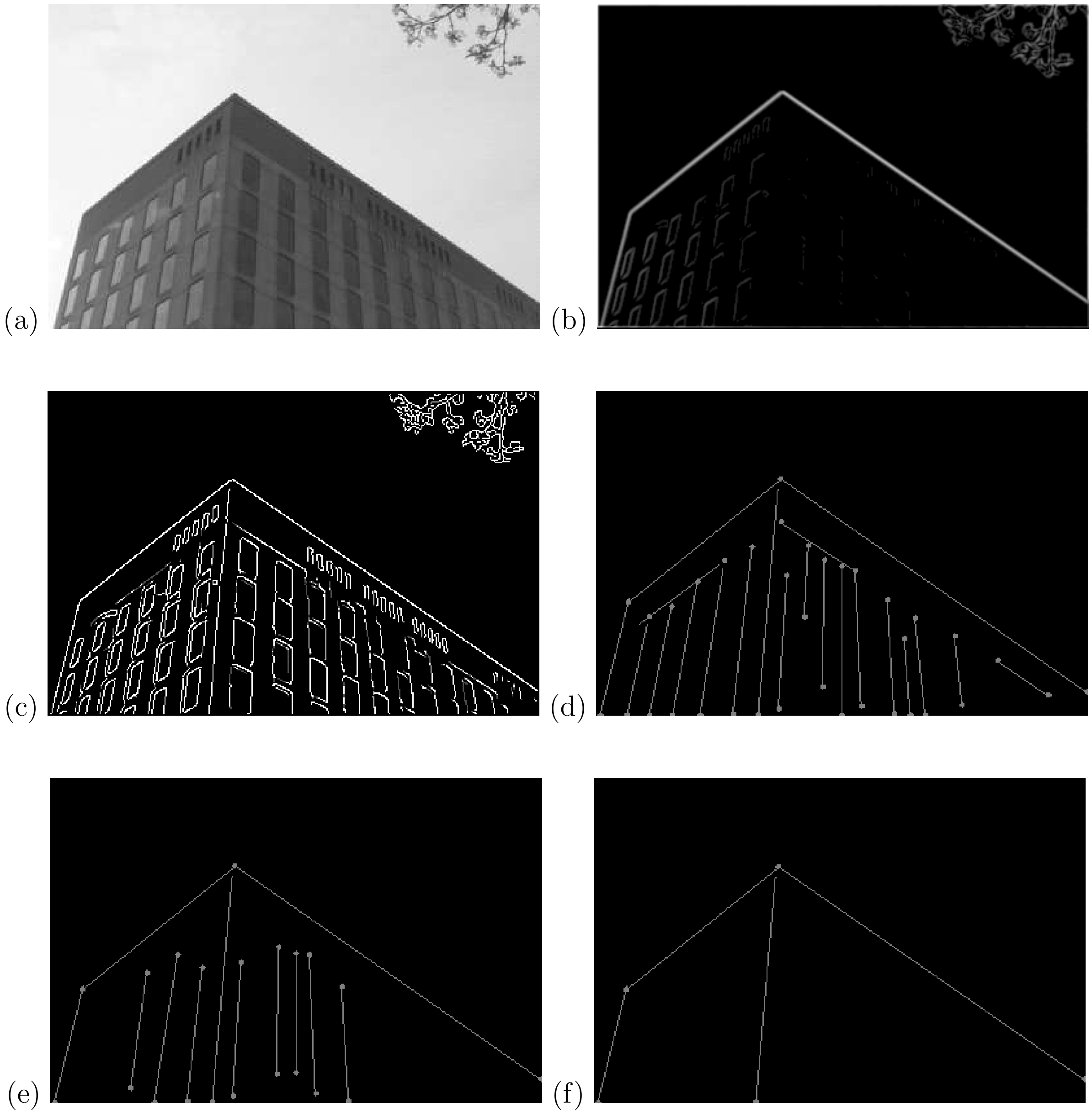
An overview of the reconstruction process is visualized in Figure 6.21. Building 545 is analyzed and recovered first, followed by building 565, and then building 575.

(a)

(b)

(c)

Figure 6.17: Three photographs from the real image data set. (a) a photograph of building 545. (b) a photograph of building 565. (c) a photograph of building 575 on the left and building 565 on the bottom right.



(a) (b) (c) (d) (e) (f)

Figure 6.18: Line segments detected for a real image with different minimum segment length threshold. (a) The original image. (b) The gradient magnitude image. (c) Canny edge output. (d) Length threshold of 60 pixels, (e) 80 pixels, (f) 100 pixels.

(a) (b) (c) (d) (e) (f)

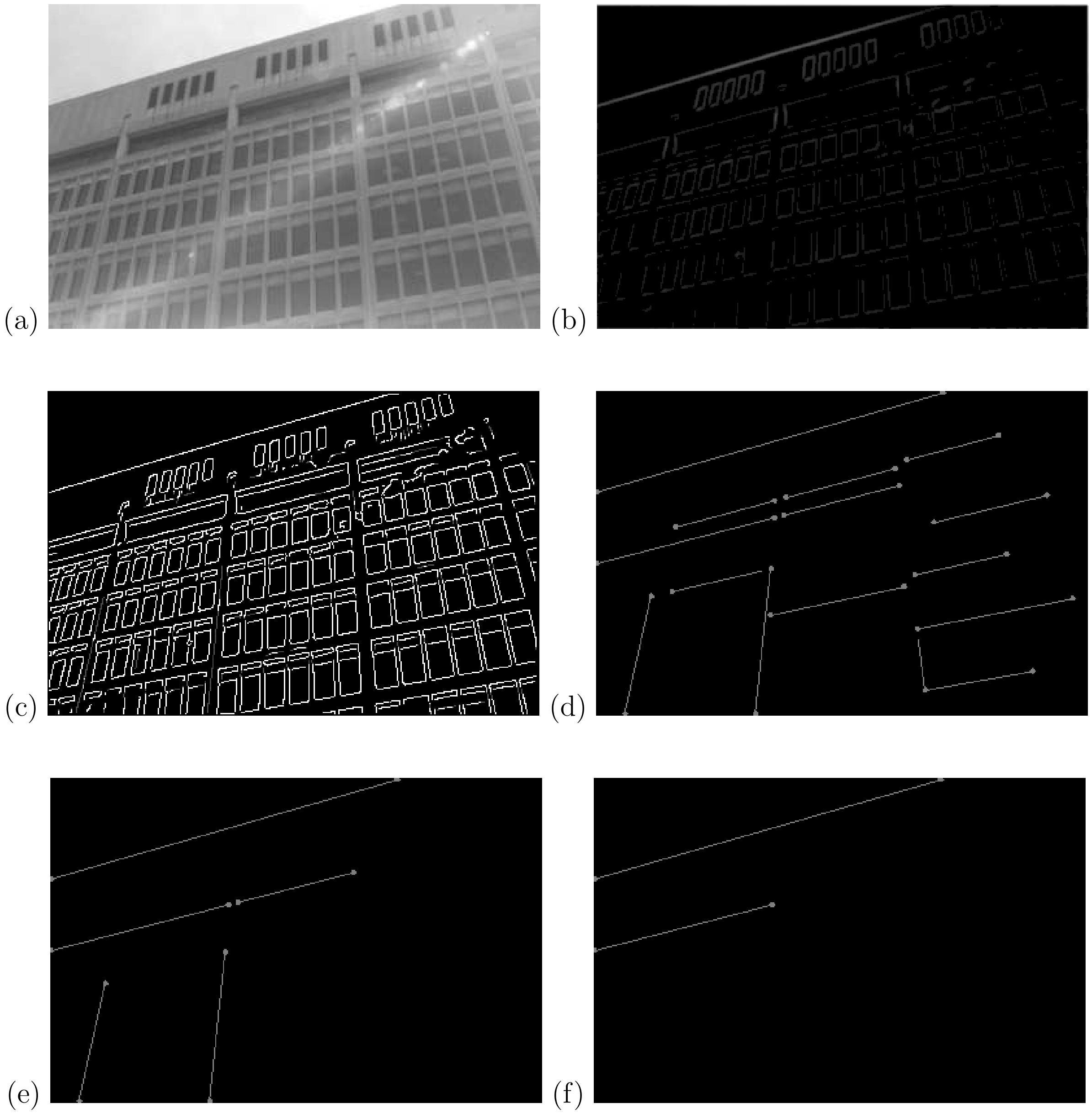


Figure 6.19: Line segments detected for a real image with different minimum segment length threshold. (a) The original image. (b) The gradient magnitude image. (c) Canny edge output. (d) Length threshold of 60 pixels, (e) 80 pixels, (f) 100 pixels.

(a) (b) (c)

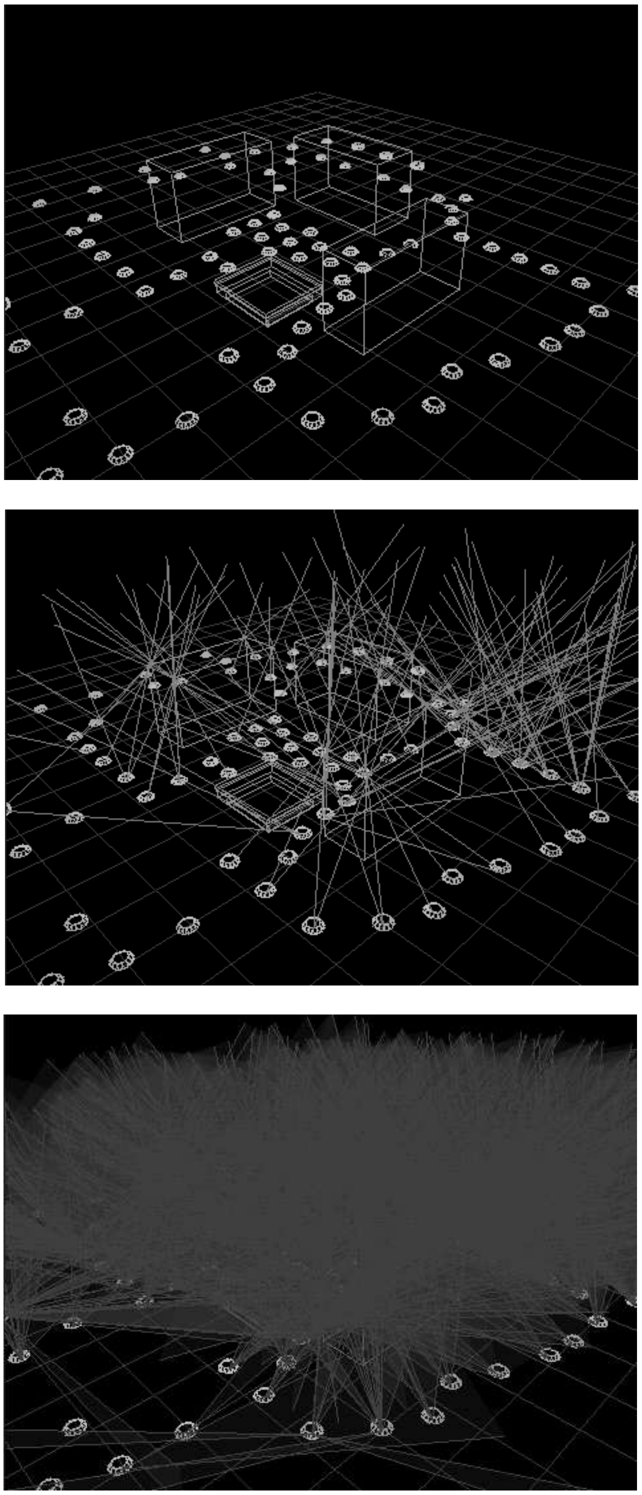


Figure 6.20: Real image reconstruction setup. (a) Image positions and building models. (b) Input vertex extrusions. (c) Input segment extrusions.

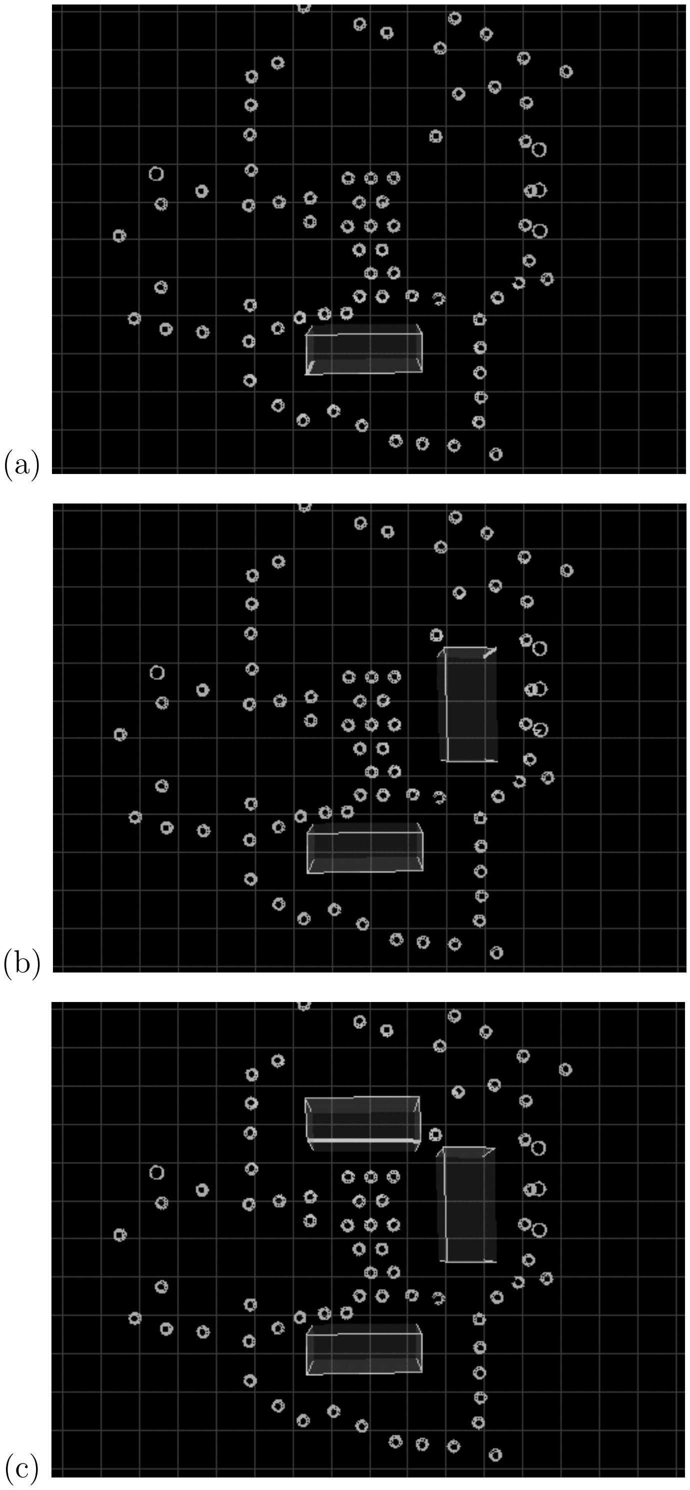


Figure 6.21: An overview of the reconstruction process. (a) Building 545 recovered. (b) Building 565 recovered. (c) Building 575 recovered.

The results presented in this section were obtained using an SGI O2 Visual Workstation running on a MIPS R10000 processor with 256MB of main memory. The average run-time for reconstructing each building is about 30 seconds.

6.3.1 Building 545 Reconstruction

Incremental reconstruction of building 545 is visualized in Figure 6.22. Figures 6.22(a),(c), and (e) show surface hypotheses, surface elements, and segment elements in different stages of reconstruction. Figures 6.22(b),(d), and (f) have segment hypotheses added to them.

The input/output relationship of the reconstruction process is displayed in Figure 6.23. The vertex extrusions and segment extrusions used as input to the system are shown in Figures 6.23(a) and (b) respectively. The output surface hypotheses, surface elements, and segment elements are shown in Figure 6.23(c).

The reconstructed building 545 is complete, with all major surfaces and segments recovered. The connections between several recovered segment elements and their corresponding extrusions are visualized in Figure 6.24.

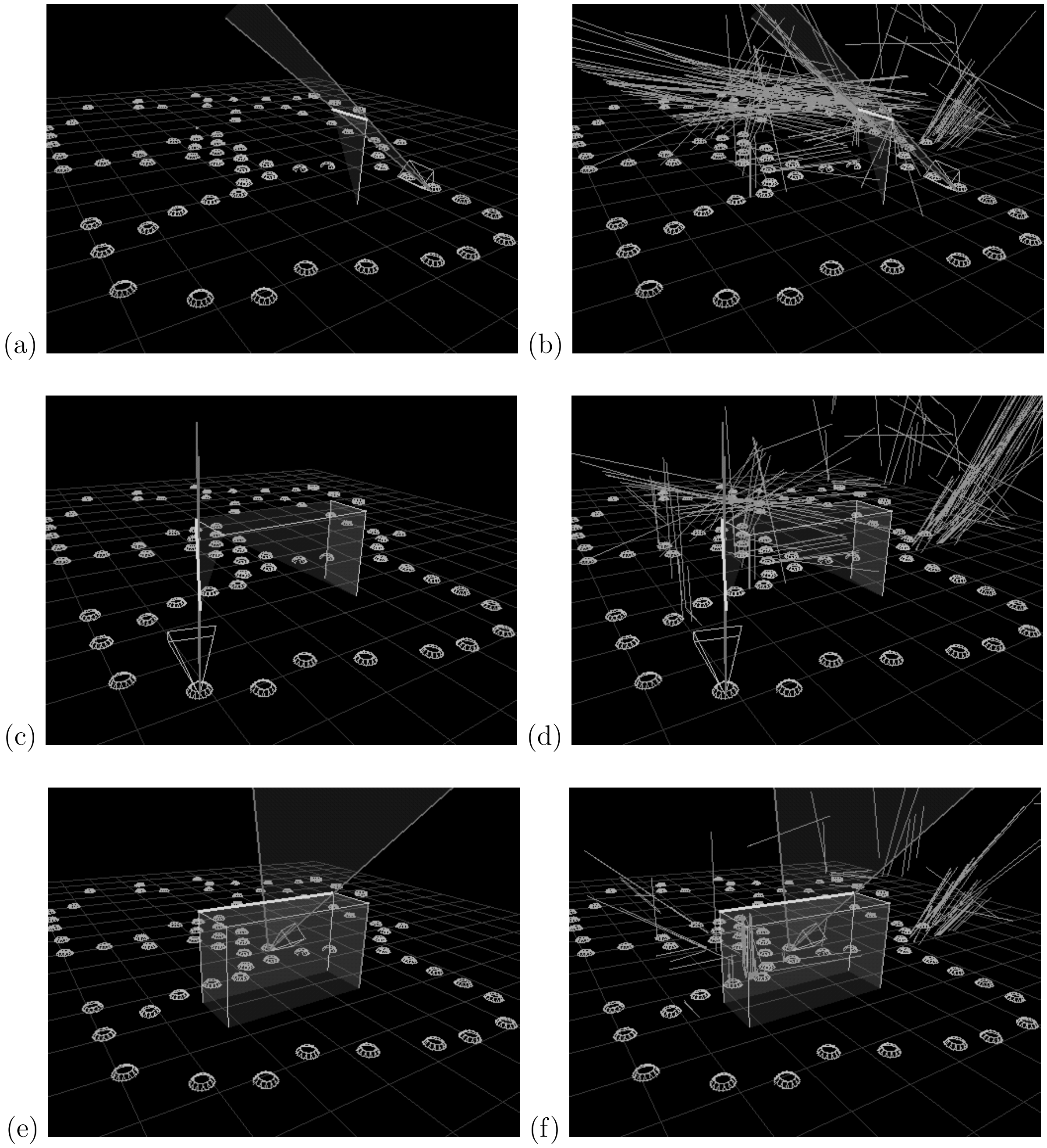
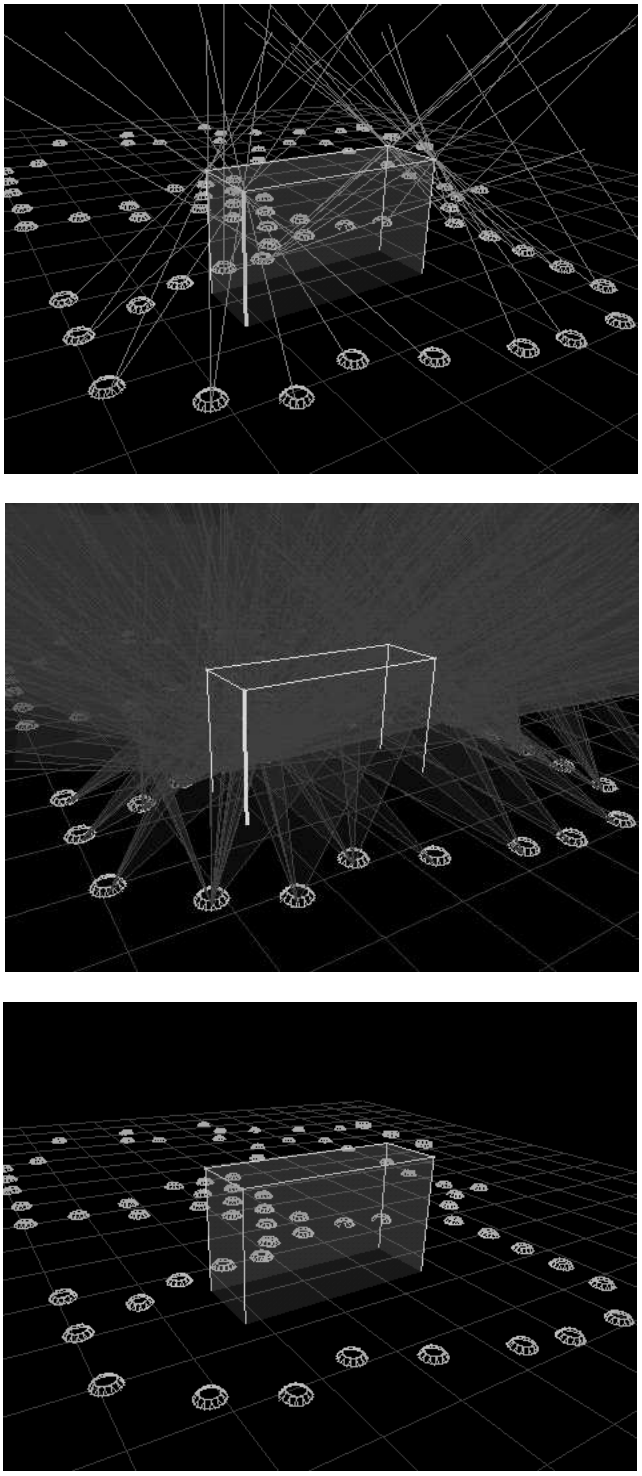


Figure 6.22: Reconstruction of building 545 in progress. (a),(c),(e) Surface hypotheses, surface elements and segment elements. (b),(d),(f) Segment hypotheses added to above.

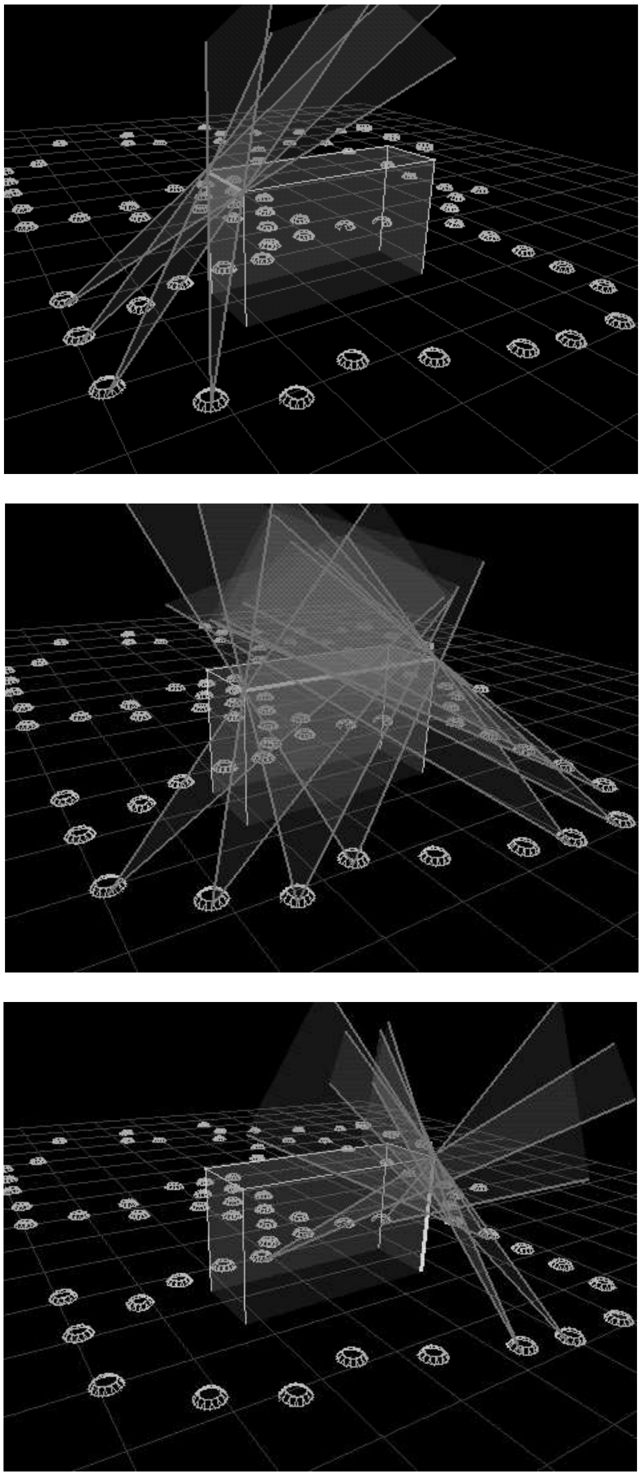


(a)

(b)

(c)

Figure 6.23: Input and output for building 545. (a) Input vertex extrusions. (b) Input segment extrusions. (c) Output surface hypotheses, surface elements, and segment elements.



(a)

(b)

(c)

Figure 6.24: Focus on recovered segments of building 545. (a) A horizontal segment. (b) A horizontal segment. (c) A vertical segment.

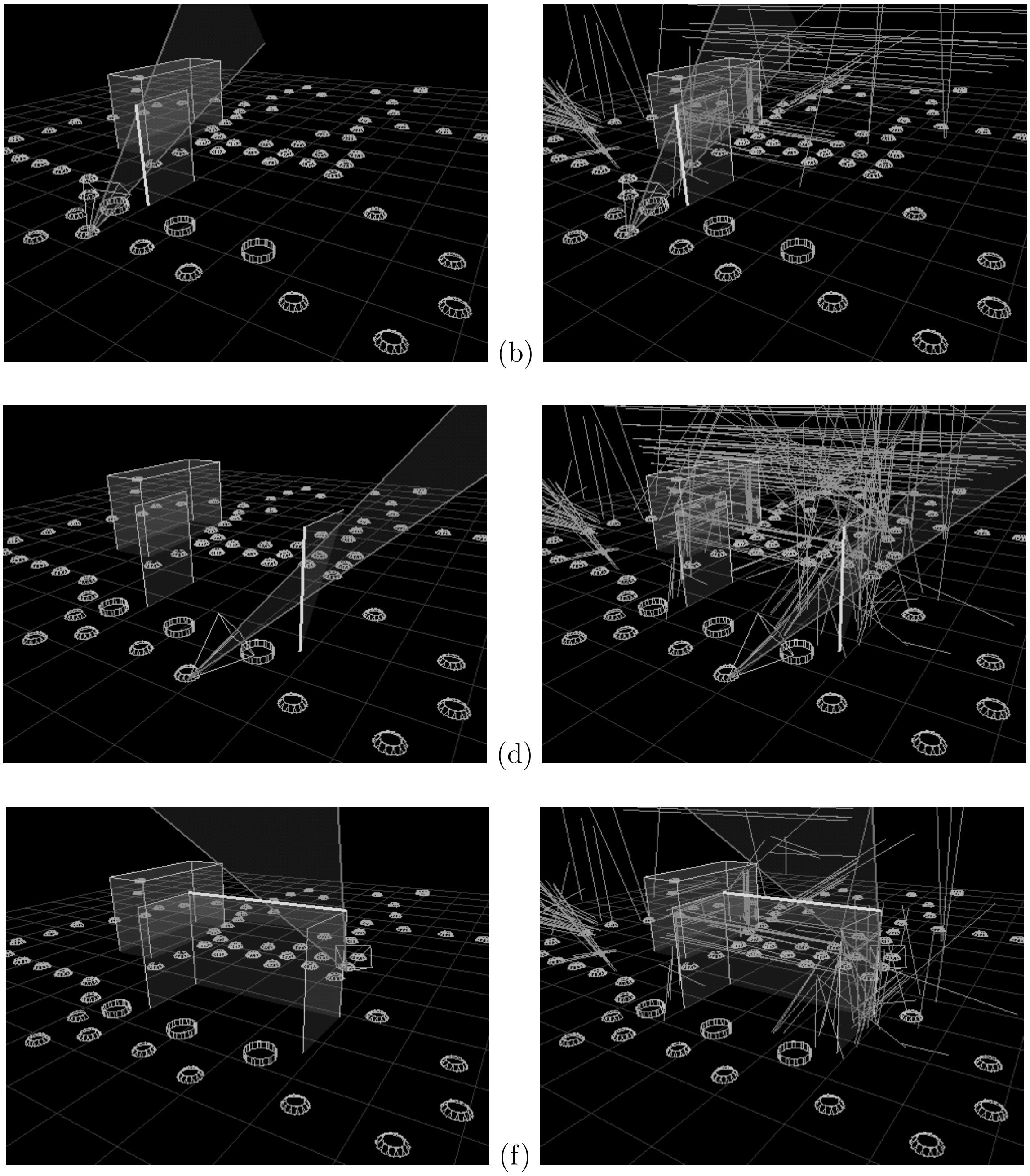
6.3.2 Building 565 Reconstruction

Incremental reconstruction of building 565 is visualized in Figure 6.25. Figures 6.25(a),(c), and (e) show surface hypotheses, surface elements, and segment elements in different stages of reconstruction. Figures 6.25(b),(d), and (f) have segment hypotheses added to them.

The input/output relationship of the reconstruction process is displayed in Figure 6.26. The vertex extrusions and segment extrusions used as input to the system are shown in Figures 6.26(a) and (b) respectively. The output surface hypotheses, surface elements, and segment elements are shown in Figure 6.26(c).

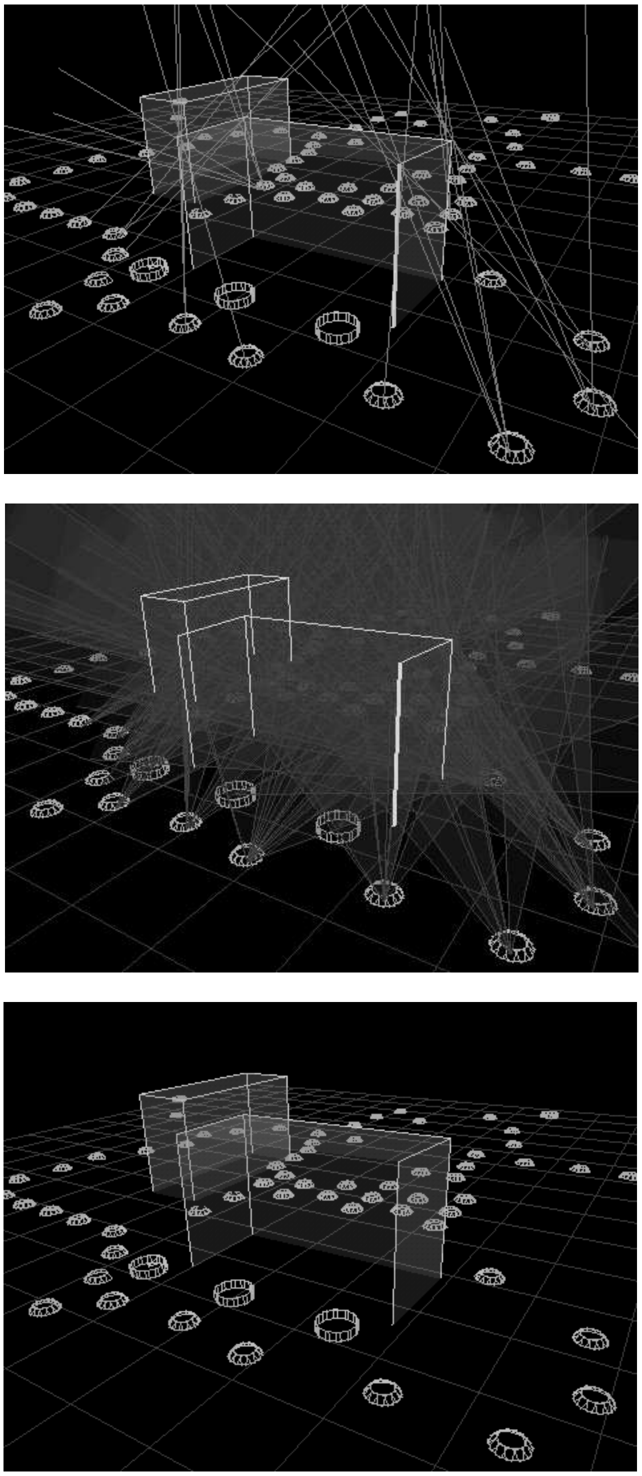
The reconstructed building 565 is incomplete, missing a line segment between the roof and the outer side surface. Figure 6.28 displays an unconfirmed segment hypothesis that is similar to the missing segment element. It shows that the segment hypothesis has incorrectly linked up segment extrusions not associated with the top edge of the building. Due to the non-convex nature of the building, an unrelated segment, close and parallel to the missing segment is visible in several of the images. The system has difficulty in distinguishing extrusions formed by the two segments.

The connections between several recovered segment elements and their corresponding extrusions are visualized in Figure 6.27.



(a) (b) (c) (d) (e) (f)

Figure 6.25: Reconstruction of building 565 in progress. (a),(c),(e) Surface hypotheses, surface elements, and segment elements. (b),(d),(f) Segment hypotheses added to above.



(a) (b) (c)

Figure 6.26: Input and output for building 565. (a) Input vertex extrusions. (b) Input segment extrusions. (c) Output surface hypotheses, surface elements, and segment elements.

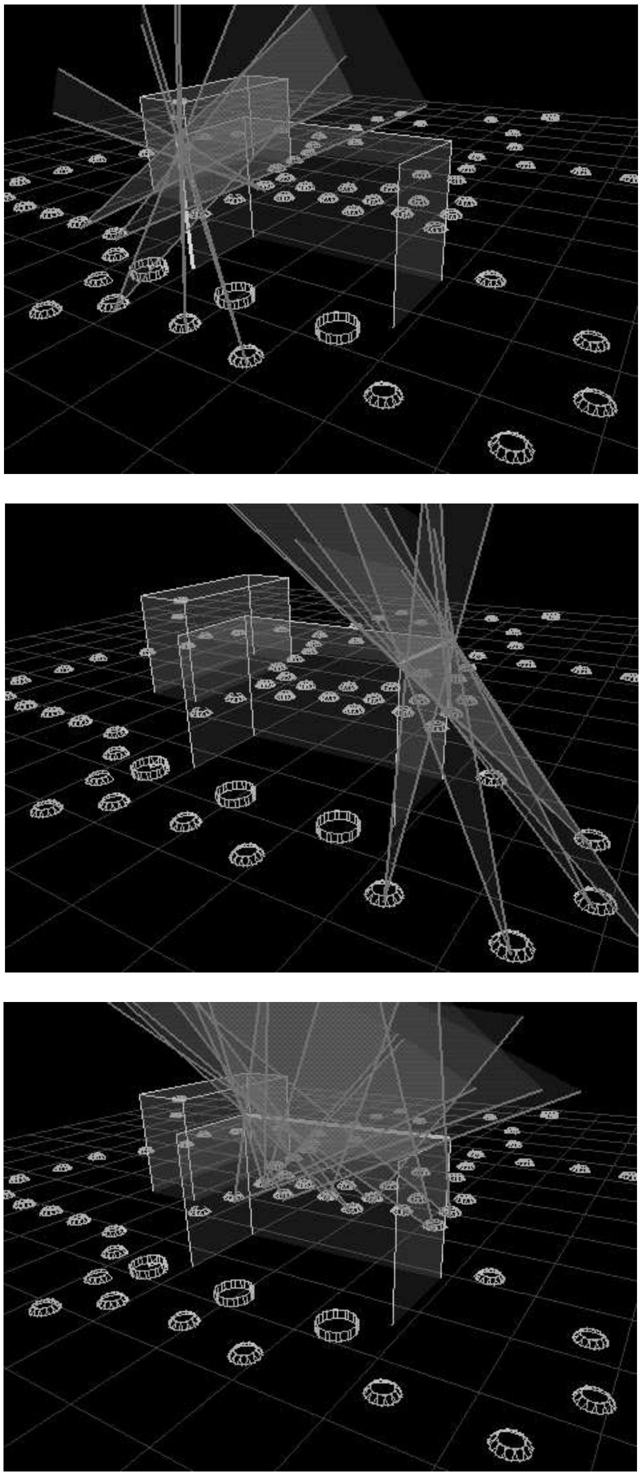

(a)
(b)
(c)

Figure 6.27: Focus on recovered segments of building 565. (a) A vertical segment. (b) A horizontal segment. (c) A horizontal segment.

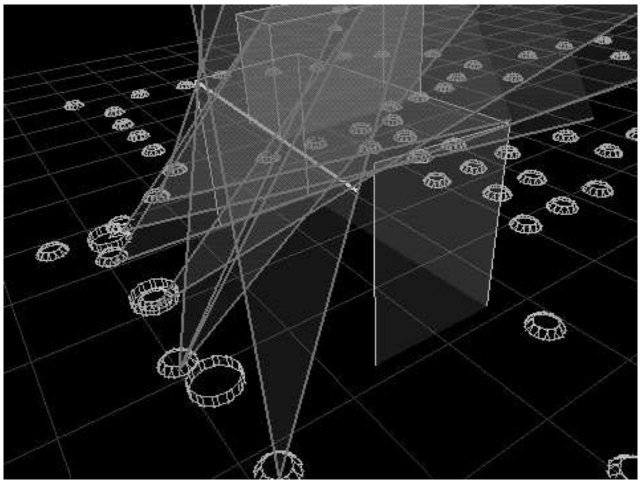


Figure 6.28: An unconfirmed segment hypothesis of building 565.

6.3.3 Building 575 Reconstruction

Incremental reconstruction of building 575 is visualized in Figure 6.29. Figures 6.29(a),(c), and (e) show surface hypotheses, elements, and segment elements in different stages of reconstruction. Figures 6.29(b),(d), and (f) have segment hypotheses added to them.

The input/output relationship of the reconstruction process is displayed in Figure 6.30. The vertex extrusions and segment extrusions used as input to the system are shown in Figures 6.30(a) and (b) respectively. The output surface hypotheses, surface elements, and segment elements are shown in Figure 6.30(c).

The reconstructed building 575 is complete, with all major surfaces and segments recovered. The connections between several recovered segment elements and their corresponding extrusions are visualized in Figure 6.31.

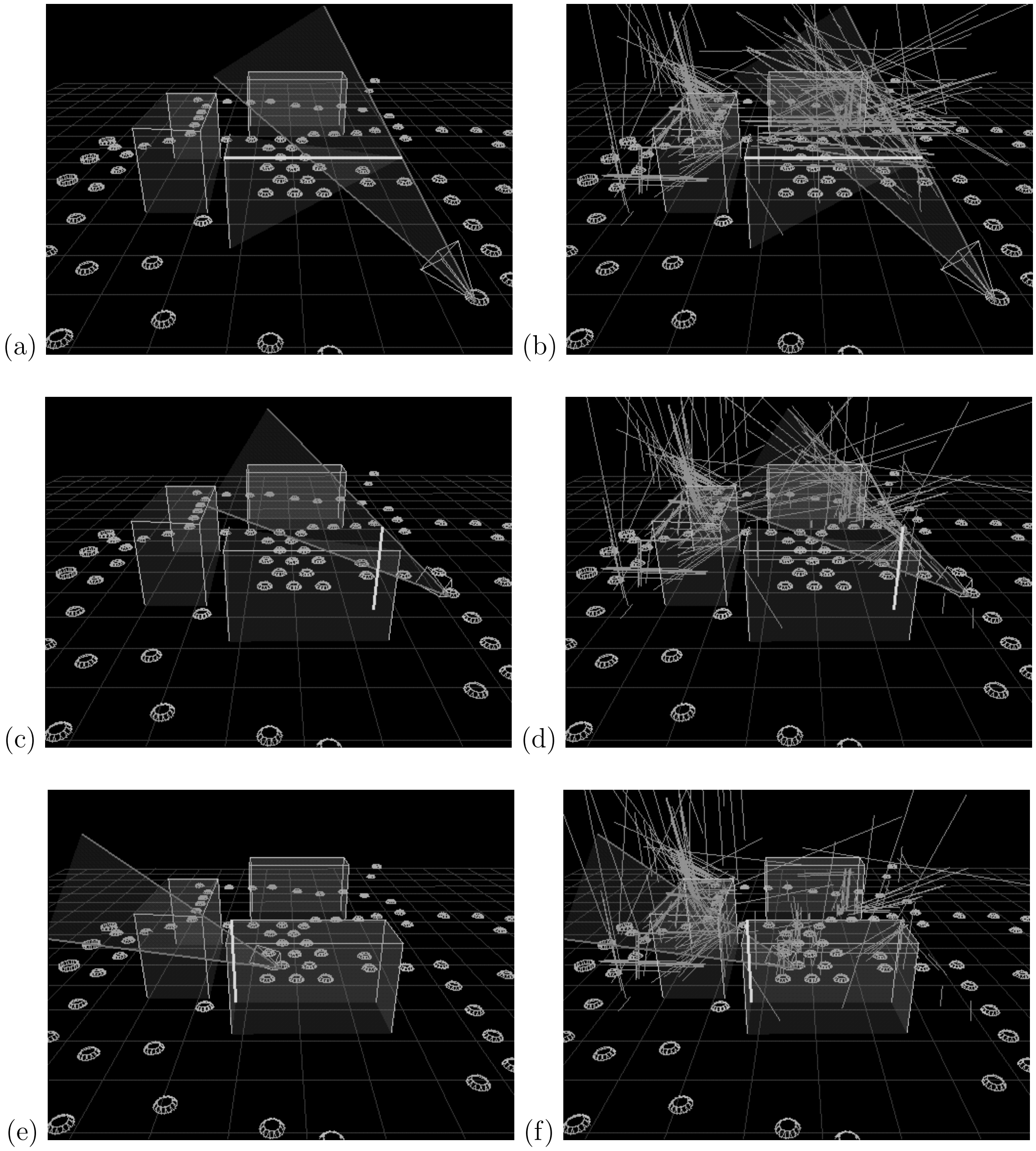
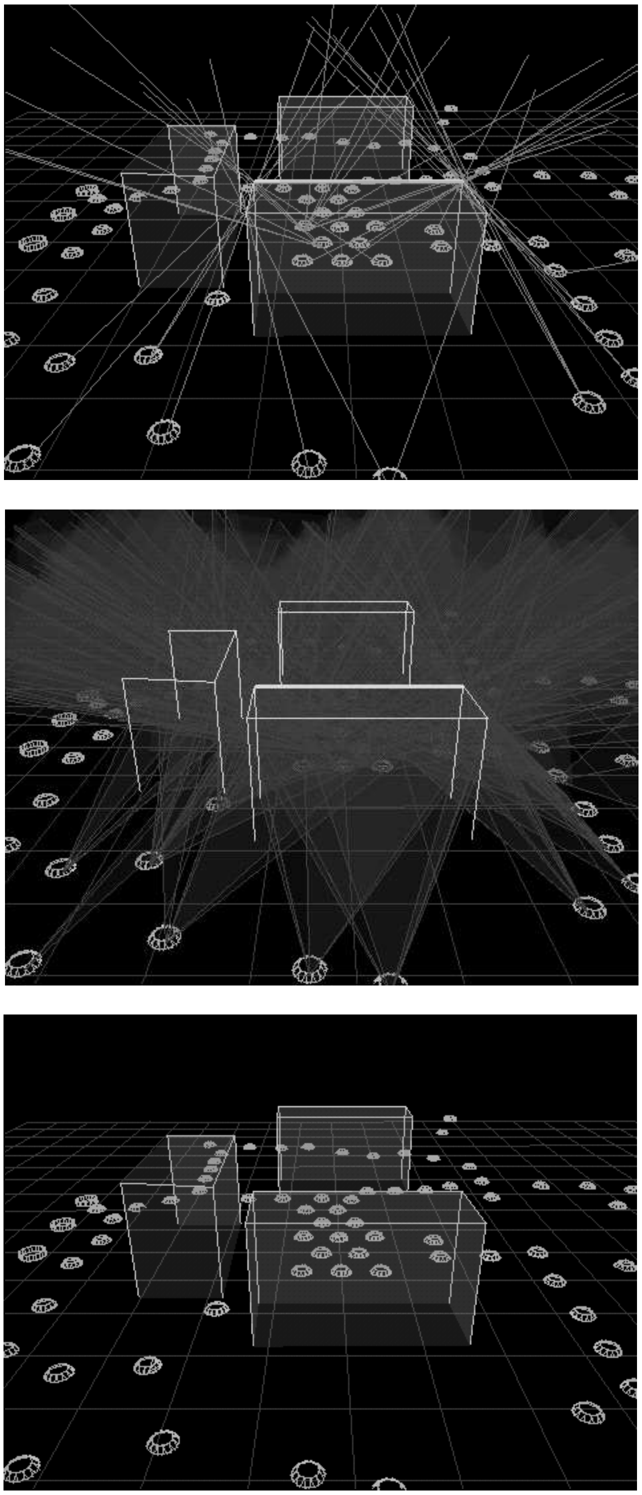


Figure 6.29: Reconstruction of building 575 in progress. (a),(c),(e) Surface hypotheses, surface elements, and segment elements. (b),(d),(f) Segment hypotheses added to above.

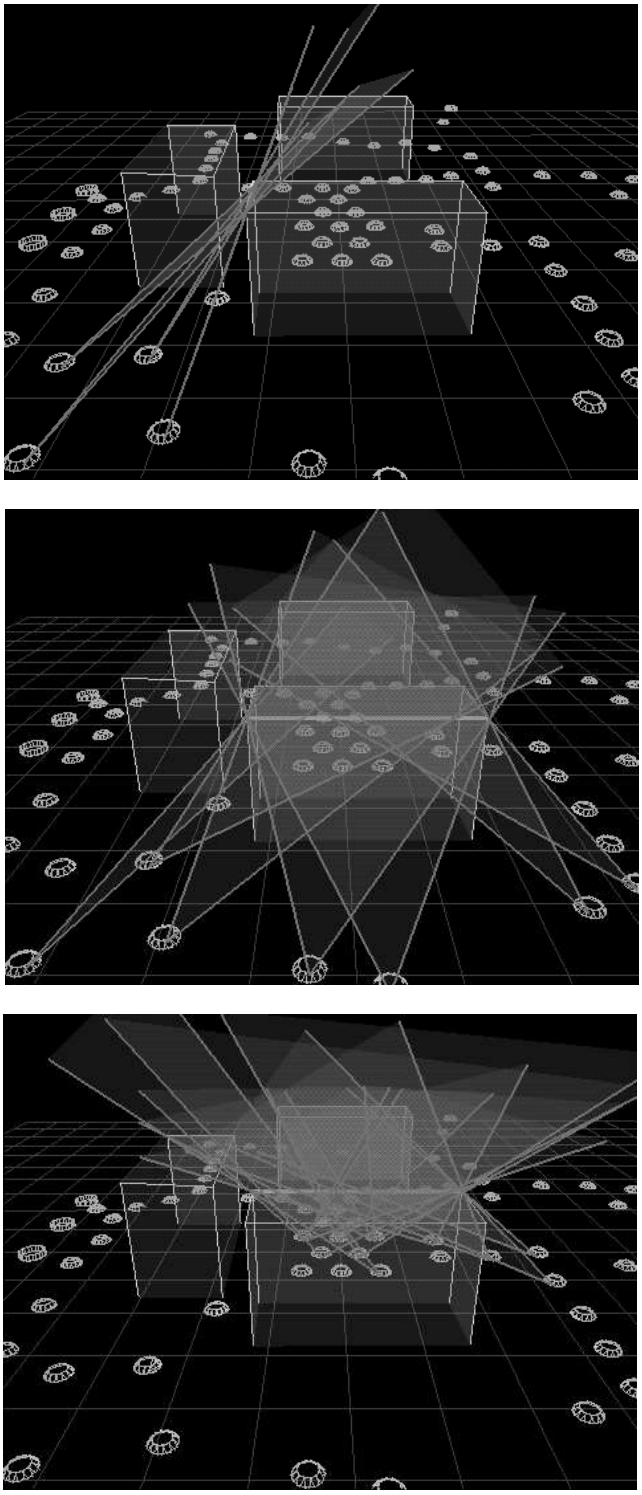


(a)

(b)

(c)

Figure 6.30: Input and output for building 575. (a) Input vertex extrusions. (b) Input segment extrusions. (c) Output surface hypotheses, surface elements, and segment elements.



(a)

(b)

(c)

Figure 6.31: Focus on recovered segments of building 575. (a) A horizontal segment. (b) A horizontal segment. (c) A horizontal segment.

6.4 Reconstruction Statistics

We have compiled reconstruction statistics on elements computed in the previous sections. Table 6.1 displays the average triangulation error and the average number of supporting features (SF) for vertex, segment, and surface elements derived from synthetic image data. Table 6.2 displays the same statistics for elements derived from real image data.

Building	Measured Averages	Vertex Elements	Segment Elements	Surface Elements
545	Triangulation error	5.55	4.26	0.43
	Number of SF	4.00	5.43	3.25
565	Triangulation error	6.28	5.10	0.00
	Number of SF	4.50	6.75	3.25
575	Triangulation error	4.70	5.62	0.54
	Number of SF	3.60	5.14	3.25
All	Triangulation error	5.56	5.00	0.32
	Number of SF	4.07	5.82	3.25

Table 6.1: Reconstruction statistics for synthetic image data

Building	Measured Averages	Vertex Elements	Segment Elements	Surface Elements
545	Triangulation error	28.08	26.52	0.30
	Number of SF	6.75	7.38	3.20
565	Triangulation error	24.23	24.58	4.26
	Number of SF	3.25	6.00	3.25
575	Triangulation error	31.16	24.43	1.83
	Number of SF	5.5	6.88	3.20
All	Triangulation error	27.82	25.20	1.98
	Number of SF	5.17	6.78	3.21

Table 6.2: Reconstruction statistics for real image data

There are several things to note in these tables:

1. The average triangulation errors are significantly higher for real image data than synthetic image data for all three types of elements. This can be attributed to the presence of noise and measurement errors in the real data.
2. The average number of supporting features are relatively similar for real and synthetic image data. This seems to indicate that the non-accidentalness probability function is the determining factor in hypothesis confirmation.
3. The average triangulation errors of surface elements are quite low for both synthetic and real image data. This is due to the fact that only a few vertex elements are involved in the formation and estimation of each surface.

6.5 Analysis of Segment Length Threshold

Changing the threshold of minimum line segment length directly affects the number of vertex and segment features generated. We have compiled statistics on the effects of changing the minimum line length threshold on the number of line segments and the reconstruction results. The data displayed in Table 6.3 represent the state of the system at the end of reconstructing building 545 using 148 real images.

Min Seg Length	No. of Seg Features	No. of Seg Extrusions	No. of Seg Hypotheses	No. of Seg Elements	Run Time (sec)
60	1050	702	2723	21	818
80	525	352	668	11	152
100	279	179	194	9	32
120	209	137	276	5	22
140	154	102	193	4	13

Table 6.3: Effects of minimum segment length threshold

Empirical observation shows that the optimal result is generated with a length threshold of 100 pixels. Higher length thresholds (120, 140 pixels) produce incomplete reconstruction. Lower length thresholds (80, 60 pixels) produce reconstruction with extra line segments.

We have plotted several pairs of parameters from the table as charts to illustrate the relationships between them:

- In Figure 6.32(a), the number of segment features detected is plotted against the minimum segment length threshold. The chart shows a rapid decline in the number of segment features as the length threshold increases.
- In Figure 6.32(b), the number of unresolved segment hypotheses remaining at the end of the reconstruction process is plotted against the minimum segment length threshold. The chart shows a even faster decline in the number of segment hypotheses as the length threshold increases, with a local minimum around 100 pixels, which produces the optimal reconstruction result.
- In Figure 6.32(c), the observed run-time of the reconstruction process is plotted against the minimum segment length threshold. The chart shows a rapid decline in the observed run-time as the length threshold increases.

- In Figure 6.33(a), the number of segment extrusions generated is plotted against the number of segment features. The dotted line shows the best-fitting line through the data points. The chart shows a strong linear relationship between the two, meaning that a constant ratio of segment features are merged in forming segment extrusions.
- In Figure 6.33(b), the number of segment elements generated is plotted against the number of segment features. The dotted line shows the best-fitting line through the data points. The chart also shows a linear relationship between the two, except for a local maximum corresponding to the length threshold of 100 pixels, which produces the optimal reconstruction result.
- In Figure 6.33(c), the observed run-time of the reconstruction process is plotted against the number of segment features. The dotted line shows the best-fitting

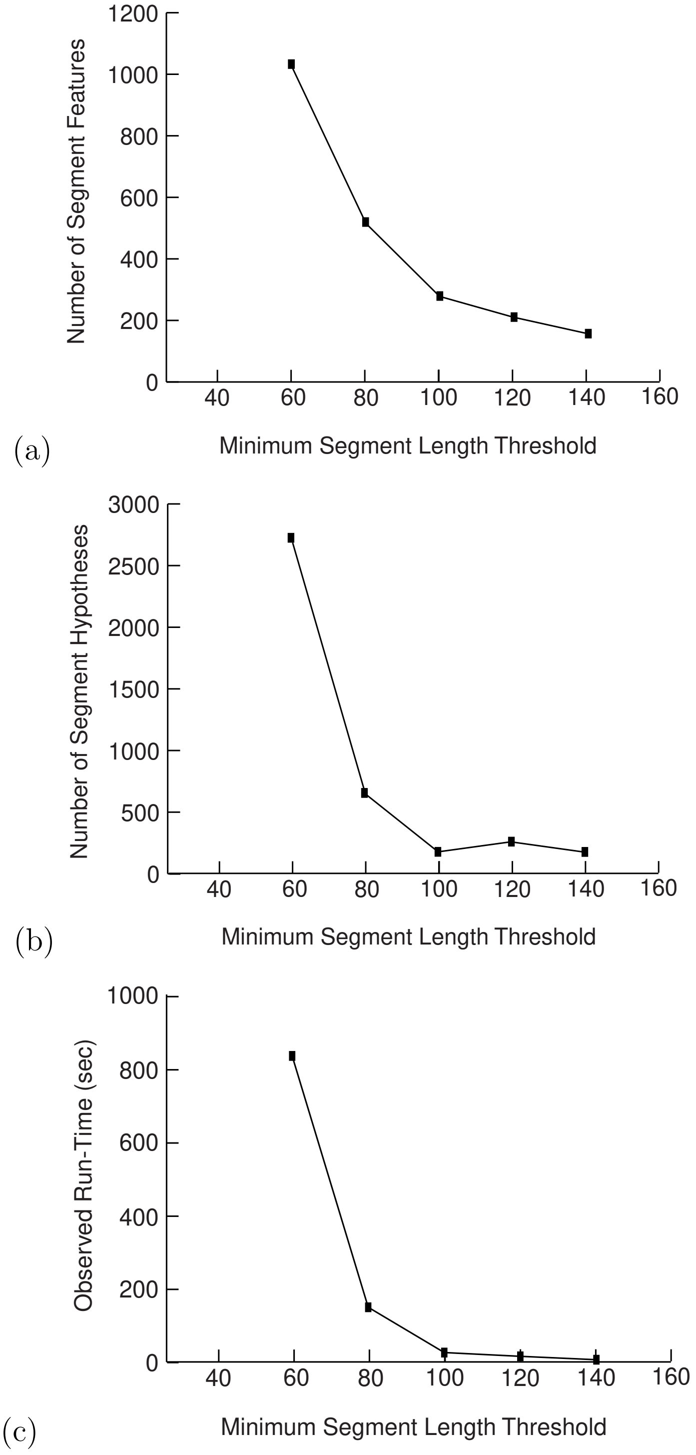
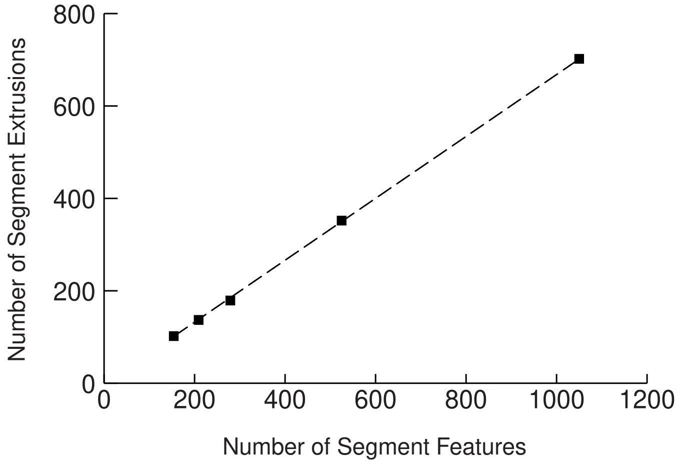
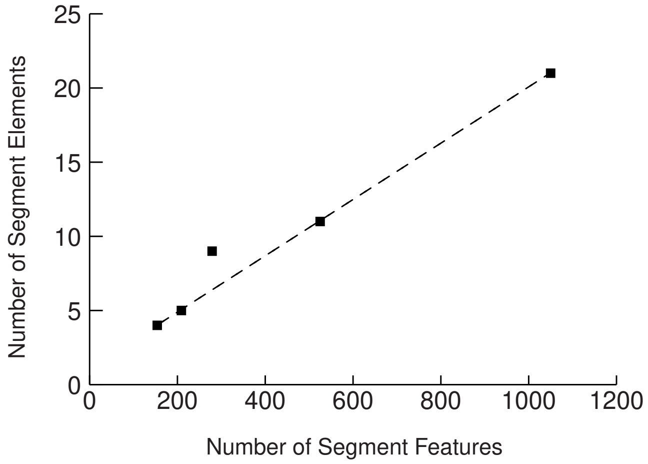


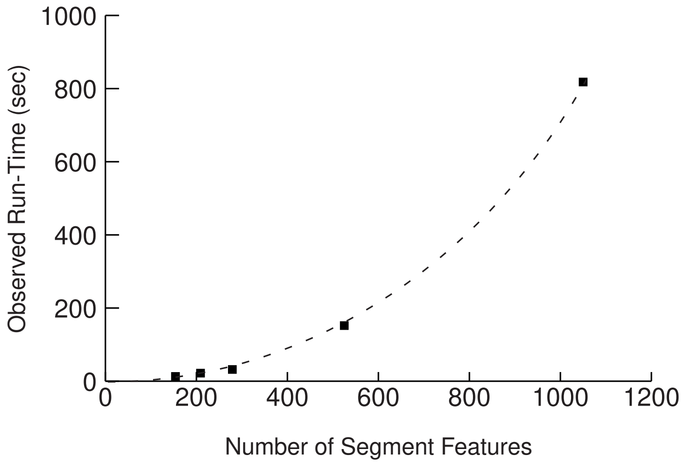
Figure 6.32: (a) Number of segment features vs minimum segment length threshold. (b) Number of unresolved segment hypotheses vs minimum segment length threshold. (c) Observed run-time in seconds vs number of segment features.



(a)



(b)



(c)

Figure 6.33: (a) Number of segment extrusions vs number of segment features. (b) Number of segment elements vs number of segment features. (c) Observed run-time in seconds vs minimum segment length threshold.

quadratic curve through the data points. The chart shows a clear quadratic relationship between the run-time and the number of input segment features. This is consistent with our knowledge that exhaustive correspondence test is performed for every pair of segment extrusions.

6.6 Analysis of Surface Visibility Constraint

In our system, we introduced the surface visible constraint as a mean to improve the low-level correspondence process. The effects of enforcing the visibility constraints of confirmed surfaces are displayed in Tables 6.4 and 6.5. Theses data represent the state of the system at the end of the reconstruction process.

Visibility Constraint	Number of Hypothesis	Hypothesis Extrusions	Extrusions per Hypothesis
Off	674	2251	3.34
On	131	493	3.76

Table 6.4: Effects of the visibility constraint on segment hypotheses

Visibility Constraint	Number of Elements	Element Extrusions	Extrusions per Element
Off	16	130	8.13
On	16	125	7.81

Table 6.5: Effects of the visibility constraint on segment elements

There are several things to note in the tables:

1. Applying the visibility constraint reduces the total number of segment hypotheses to 131 from 674. Many mis-matches that violate the visibility constraint are no longer retained as hypotheses.

2. Applying the visibility constraint reduces the average number of supporting Extrusions per Element (EpE) to 7.81 from 8.13. This is consistent with the clipping effect of the visibility constraint.

3. Applying the visibility constraint increases the average number of supporting Extrusions per Hypothesis (EpH) to 3.76 from 3.34. Contrast this with the effect on elements (EpE), which was decreased. The key here is that mis-matches which violate the visibility constraint tend to have fewer supporting extrusions than normal hypotheses. When they are eliminated from the pool of hypotheses, EpH is raised as a result.

6.7 Summary

In this chapter, we have accomplished the following tasks:

1. We described the graphical user interface of the reconstruction system, covering the main image window, the virtual world window, and the reconstruction control panel.

2. We presented reconstruction results of our system operating on the synthetic image data set of the Technology Square. The results for buildings 545, 565, and 575 are reasonable, except for areas where there are insufficient image coverage.

3. We presented reconstruction results of our system operating on the real photograph data set of the Technology Square. The results for buildings 545, and 575 are reasonable, but not so for building 565 which was hampered by difficult feature detection and non-convex building structure.

4. We compiled statistics showing that the average triangulation errors are significantly higher for real image data than synthetic image data due to noise and error in measurements, and that the average number of supporting features are relatively similar for real and synthetic image data.

5. We ran multiple experiments with different segment length thresholds. Charts were then plotted illustrating that as the segment length threshold is increased, the number of segment features generated decreases rapidly, and that the observed run-time of the reconstruction process is approximately a quadratic function of the number of input segment features.

6. We compiled statistics demonstrating that the visibility constraints of confirmed surfaces play a significant role in improving the correspondence process by eliminating many accidental mis-matches.

Chapter 7

Conclusion

In this thesis, we set out to design and implement a large-scale 3D reconstruction system with the capabilities of performing correspondence in 3D, incremental processing, and surface generation. These capabilities, we believe, are fundamental to all future computerized systems seeking to solve more complex reconstruction tasks. The arguments for our belief have been presented in Chapter 1.

After we outlined our strategy, we described the structure of our correspondence system in Chapter 2. At the heart of our system are the triangulation methods for intersecting feature extrusions, and they are developed in Chapter 3. The evaluation and processing of correspondence hypotheses are presented next in Chapter 4. Once segment hypotheses have been confirmed, surface computation will be activated, as described in Chapter 5.

The reconstruction results of Chapter 6 demonstrated that we have achieved our initial objectives, for the prescribed input and testing conditions. Although our experiments are limited in nature, they do provide insights into our reconstruction system. We shall discuss them briefly in this chapter.

7.1 What We Learned

Testing our system on synthetic images and real photographs have given us valuable feedback about our approach and our system. Looking back, we can see what works for us, and what are the challenges that remain.

What works for us include:

- Triangulation-based method – The ability to establish feature correspondence in 3D is critical for being able to handle significant viewpoint changes in the input images. The method allows us to overcome the dual problem of solving for the correspondence problem and 3D structure *simultaneously.*

- Hypothesis-based approach – In accordance to Marr's principle of least committment, our hypothesis-based approach gradually filters the good hypotheses from the bad ones. Although our exhaustive approach to hypothesis generation is burdensome, it ensures that no valid hypothesis will be overlooked.

- Multi-level processing – By combining both bottom-up extrusion triangulation information with top-down constraints of confirmed surfaces, better decisions can be made in regard to uncertain feature correspondence hypotheses. Higher-levels of scene representation could potentially offer more useful constraints.

The challenges that remain for us are:

- Inflexible built-in assumptions – Our assumptions of linear building contours, planar and opaque surfaces, and convex building structure have made our system rather inflexible. These assumptions serve to greatly reduce the complexities of our system. But, when faced with deviations from these assumptions, the output of the system can become unpredictable.

- Unreliable feature detection – By adopting a hypothesis-based approach, our system is less vulnerable to unreliable feature detection. Nonetheless, the system is affected by multiple incorrect and missed features. In feature detection, the source of problems appears to be our use of thresholds.

- Arbitrary image ordering – Currently, the images are inserted into the system in an arbitrary order, as long as they satisfy certain spatial constraints. We note that the ordering of input images may either accelerate or delay the confirmation of valid correspondence hypotheses. If the input can be arranged so that valid hypotheses are confirmed sooner, a more efficient system would result.

The successes listed above represent the main contributions of this thesis to the problem of 3D reconstruction. At the same time, the challenges represent interesting areas for future research and development.

7.2 Theoretical Connections

On the surface, the correspondence problem in 3D reconstruction appears to be a very specific problem in computer vision. However, as we will see, there are many interesting theoretical connections between this problem, and problems in other fields.

Recall in Chapter 1 we mentioned that in 3D reconstruction we face the dual problem of solving for the feature correspondence and 3D structure *simultaneously*. This dual problem, may be viewed as a multi-dimensional data clustering problem, involving simultaneous parameter estimation and data labeling. The estimation process determines the parameters of the 3D structure of interest, while the data labeling process groups observations from similar sources. The two processes are tightly interlocked together; each depending on the other process for input information.

In pattern recognition, the traditional solution to the data clustering problem is the K-means clustering algorithm [20]. It is a hypothesis-based method which begins by hypothesizing a certain clustering relationship. Assuming the cluster assignment is true, the centroid for each cluster is computed. Next, one performs optimal (minimum distance) classification, and updates all of the clusters. These steps iterate until some convergence criterion is met. In neural computation, this is also known as an unsupervised competitive learning algorithm [27].

Closely related to data clustering is the work in vector quantization. Vector quantization is a data compression technique that maps real vectors into discrete symbols. The K-means algorithm, also known as the Lloyd algorithm, is used to determine the codebook. At each stage of the Lloyd algorithm, the expected distortion is decreased until the algorithm converges to a local minimum of distortion [15]. The algorithm partitions the input space into distinct regions, which represent the codebook. Vector quantization has been successful applied to the field of speech recognition [41].

The connection to data compression is interesting because in a typical reconstruction project, the amount of redundancy in image data can be very high. A small number of underlying structures may be responsible for a large amount of image data. Therefore, it is efficient to describe the scene in terms of the underlying 3D models and a series of camera poses. Data compression could be formulated as an objective for global optimization in 3D reconstruction. We note that minimum description length (MDL) criterion has been successfully applied to image segmentation [30], [17]. The method finds the simplest image description in terms of description length, and is closely related to the maximum likelihood and MAP criteria.

The iterative process of the K-means algorithm also resembles the EM (expectation maximization) algorithm in Bayesian statistics. The EM algorithm is an iterative method for finding the local maximum of the posterior density, starting from some initial guess [19], [50]. Each iteration consists of an E-step, which finds the expectation of the needed functions (sufficient statistics) of the missing data, using observed values and hypothesized parameters, and a M-step, which estimates the parameters by maximizing the needed functions found in the E-step. These two steps are then repeated until convergence.

It has been shown that the EM algorithm is suitable for solving the data clustering problem in the context of mixture estimation [42], [29]. This suggests that the reconstruction problem may also be solved by an EM-based approach. The key idea behind EM is that augmentation of the observed data with missing data can simplify the parameter estimation process [23]. For information that are not observable, such as the unknown correspondence relationship in our case, suitable probabilistic models for filling in the missing information would be applied. Conceivably, this could be more efficient than our exhaustive approach to generating new correspondence hypotheses.

7.3 Summary

In Chapter 1, we considered the problem of large-scale 3D reconstruction and the design of a suitable system. We first surveyed existing work in 3D reconstruction. We then discussed the challenges of large-scale 3D reconstruction, arguing that any system designed to tackle the problem should have these three capabilities: correspondence in

3D, incremental processing, and surface generation. We then motivated a triangulation-based approach to solving the dual problem of feature correspondence and structure estimation.

In Chapter 2, we provided an overview of our correspondence system. We first examined the correspondence problem in the context of a triangulation-based approach. Using two examples, we demonstrated the necessity of treating all initial matches as hypotheses that must be confirmed by additional evidence. We then proceeded to define the main data structures used in our system, and outlined the correspondence algorithm with a flow diagram.

In Chapter 3, we presented our triangulation methods for correspondence. We began by describing the hypothesis formation process that leads to triangulation. Then we derived two multi-image triangulation methods for computing the intersections of vertex extrusions and segment extrusions in 3D space. The methods were developed in a probabilistic framework to highlight the uncertainties involved. Special situations under which these methods may fail were also noted.

In Chapter 4, we examined the processing of correspondence hypotheses. The computation of posterior hypothesis probability involves two components: the data consistency probability, which is a function of triangulation residual, and the non-accidentalness probability, which is a function of the number of supporting features. Based on posterior probability, a hypothesis may either be confirmed, updated, or rejected. We outlined the chain of events that occurs in each case.

In Chapter 5, we described the computation of surfaces in our system. The two major tasks involved in surface computation are: surface hypothesis processing, and enforcement of the visibility constraints of confirmed surfaces. These tasks are accomplished under the assumptions that the surfaces of interest can be effectively modeled by piece-wise planar and opaque polygons. Maximum likelihood estimators for the parameters of planar surfaces were derived.

In Chapter 6, we presented the results of our system in reconstructing the Technology Square, first using synthetic image data, and then real image data. Computed statistics showed that reconstruction results from real images have much higher average triangulation errors than results from synthetic images. Next, we ran experiments which demonstrated that the observed run-time of the reconstruction process is approximately a quadratic function of the number of input segment features. Finally, our analysis showed the visibility constraints of confirmed surfaces play a significant role in improving the correspondence process.

Appendix A

Default Parameters

The default parameter settings for our reconstruction system are listed in this appendix.

A.1 Edge Detection Parameters

- Gradient magnitude threshold (edge_flags.abs) = 20.0.
- Upper threshold ratio (edge_flags.high) = 0.7.
- Lower threshold ratio (edge_flags.low) = 0.3.
- RGB red coefficient (edge_flags.red) = 0.5.
- RGB green coefficient (edge_flags.green) = 0.3.
- RGB blue coefficient (edge_flags.blue) = 0.2.
- Gaussian filter scale factor (edge_flags.sigma) = 1.0.
- Gaussian filter residue (edge_flags.residue) = 0.05.

A.2 Line Extraction Parameters

- Hough space radial dimension (line_flags.rdim) = image diagonal.
- Hough space angular dimension (line_flags.tdim) = 720.
- Hough transform threshold (line_flags.hough) = 20.

- Minimum line length without junction (line_flags.length1) = 100.0.
- Minimum line length with junction (line_flags.length2) = 30.0.
- Maximum angular difference (line_flags.angle) = 20.0.
- Hough transform angular window (line_flags.awin) = 2.
- Hough transform radial window (line_flags.rwin) = 2.
- Hough transform center window (line_flags.cwin) = 1.
- Maximum line gap (line_flags.max_gap) = 3.
- Maximum line free end length (line_flags.free_len) = 30.
- Maximum line intersection error (line_flags.error) = 3.0.

A.3 Vertex Extraction Parameters

- Maximum gap size (vert_flags.max_gap) = 10.
- Maximum vertex overlap (vert_flags.level) = 5.
- Minimum angular difference (vert_flags.angle) = 15.0.
- Minimum incident line length (vert_flags.min_len) = 20.0.

A.4 3D Reconstruction Parameters

- Maximum extrusion merge angle (recon.merge_angle) = .035.
- Minimum number of features (recon.min_nfeat) = 3.
- Maximum VXH error (recon.VXH_error) = 100.0.
- Maximum VXX error (recon.VXX_error) = 40.0;
- Maximum visual angle (recon.visual_angle) = .4363.
- Maximum vertical slope (recon.vert_slope) = .1.

- Maximum LXE error (recon.LXE_error) = 100.0.
- Maximum LXH error (recon.LXH_error) = 10.0.
- Maximum LXX error (recon.LXX_error) = 100.0.
- Minimum LXX angle (recon.LXX_angle) = .2000.
- Default ground level (recon.ground_level) = 300.0.
- Maximum surface merge angle (recon.merge_angle2) = .09.
- Maximum line-plane distance (recon.plane_dist) = 15.0.
- Maximum line-plane angle (recon.plane_angle) = .09.
- Vertex hypothesis confirmation probability (recon.pc_VH) = 0.5.
- Vertex hypothesis rejection probability (recon.pr_VH) = 0.2.
- Segment hypothesis confirmation probability (recon.pc_LH) = 0.5.
- Segment hypothesis rejection probability (recon.pr_LH) = 0.2.
- Vertex data consistency probability σ (recon.ds0_VH) = 20.0.
- Vertex data consistency probability φ (recon.dv0_VH) = 0.5.
- Vertex data consistency probability $\mathcal{E}$ (recon.de0_VH) = 1.0×10^8.
- Segment data consistency probability σ (recon.ds0_LH) = 10.0.
- Segment data consistency probability φ (recon.dv0_LH) = 0.5.
- Segment data consistency probability $\mathcal{E}$ (recon.de0_LH) = 1.0×10^8.
- Vertex non-accidentalness probability φ (recon.nv0_VH) = 0.1.
- Vertex non-accidentalness probability p_0 (recon.np0_VH) = 0.1.
- Vertex non-accidentalness probability p_1 (recon.np1_VH) = 0.5.
- Segment non-accidentalness probability φ (recon.nv0_LH) = 0.1.

- Segment non-accidentalness probability p_0 (recon.np0_LH) = 0.1.
- Segment non-accidentalness probability p_1 (recon.np1_LH) = 0.5.

Bibliography

[1] Adiv, G., "Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7, p384-401, 1985.

[2] Antone, M. and Teller, S., "Automatic Recovery of Relative Camera Rotations for Urban Scenes", *Proc. Conf. on Computer Vision and Pattern Recognition*, p282-289, 2000.

[3] Ayache, N., and Faverjon, B., "Efficient Registration of Stereo Images by Matching Graph Descriptions of Edge Segments", *Intl. J. of Computer Vision*, 2, p107-131, 1987.

[4] Bar-Shalom, Y., and Fortmann, T.E., *Tracking and Data Association*, Academic Press, Orlando, FL, 1988.

[5] Baker, H.H., and Bolles, R.C., "Generalizing Epipolar-Plane Image Analysis on the Spatiotemporal Surface", *Intl. J. of Computer Vision*, 3, p33-49, 1989.

[6] Bolles, R.C., Baker, H.H., and Marimont, D.H., "Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion ", *Intl. J. of Computer Vision*, 1, p7-55, 1987.

[7] Boyer, K.L., and Kak, A.C., "Structural Stereopsis for 3-D Vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10, p144-166, 1989.

[8] Bedekar, A.S., and Haralick, R.M., "Finding Corresponding Points Based on Bayesian Triangulation", *Proc. Conf. on Computer Vision and Pattern Recognition*, p61-66, 1996.

[9] Chou, G.T. and Teller, S., "Multi-Image Correspondence using Geometric and Structural Constraints", *Proc. Image Understanding Workshop*, p794-808, 1997.

[10] Chou, G.T. and Teller, S., "Multi-Level 3D Reconstruction with Visibility Constraints", *Proc. Image Understanding Workshop*, p543-550, 1998.

[11] Chung, R. and Nevatia, R., "Use of Monocular Groupings and Occlusion Analysis in a Hierarchical Stereo System", *Computer Vision and Image Understanding*, 62(2), p245-268, 1995.

[12] Collins, R.T., "A Space-Sweep Approach to True Multi-Image Matching", *Proc. Conf. on Computer Vision and Pattern Recognition*, p358-363, 1996.

[13] Coorg, S., Master, N., and Teller, S., "Acquisition of a Large Pose-Mosaic Dataset", *Proc. Conf. on Computer Vision and Pattern Recognition*, p872-878, 1998.

[14] Coorg, S., and Teller, S., "Extracting Textured Vertical Facades from Controlled Close-Range Imagery", *Proc. Conf. on Computer Vision and Pattern Recognition*, p625-632, 1999.

[15] Cover, T.M., and Thomas, J.A., *Elements of Information Theory*, John Wiley & Sons, Inc., New York, NY, 1991.

[16] Crowley, J.L., Stelmaszyk, P., Skordas, T., and Puget, P., "Measurement and Integration of 3D Structures by Tracking Edge Lines", *Intl. J. of Computer Vision*, 8, p29-52, 1992.

[17] Darrell, T., Sclaroff, S., and Pentland, A.P., "Segmentation by Minimal Description", *Proc. 3rd Intl. Conf. on Computer Vision*, p173-177, 1990.

[18] Debevec, P.E., Taylor, C.J., and Malik, J., "Modeling and Rendering Architecture from Photographs", *Proc. ACM SIGGRAPH*, p11-20, 1996.

[19] Dempster, A., Laird, N., and Rubin, D., "Maximum Likelihood for Incomplete Data via the EM Algorithm", *J. of the Royal Statistical Society*, 39, p1-38, 1977.

[20] Duda, R.O., and Hart, P.E., *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.

[21] Faugeras, O., Lustman, F., and Toscani, G., "Motion and Structure from Motion from Point and Line Matches", *Proc. 1st Intl. Conf. on Computer Vision*, p25-34, 1987.

[22] Faugeras, O., *Three-Dimensional Computer Vision*, The MIT Press, Cambridge, MA, 1993.

[23] Gelman, A., Carlin, J., Stern, H., and Rubin, D., *Bayesian Data Analysis*, Chapman & Hall, London, UK, 1995.

[24] Grimson, W.E.L., "A Computer Implementation of a Theory of Human Stereo Vision", *Phil. Trans. Royal Soc. London*, B292, p217-253, 1981.

[25] Heeger, D.J., and Jepson, A.D., "Subspace Methods for Recovering Rigid Motion", *Intl. J. of Computer Vision*, 7, p95-117, 1992.

[26] Herman, M., and Kanade, T., "The 3D Mosaic Scene Understanding System", *Artificial Intelligence*, 30, p289-341, 1986.

[27] Hertz, J., Krogh, A., and Palmer, R.G., *Introduction to the Theory of Neural Computation*, Addison-Wesley Publishing Company, Redwood City, CA, 1991.

[28] Horaud, R., and Skordas, T., "Stereo Correspondence Through Feature Groupings and Maximal Cliques", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, p1168-1180, 1989.

[29] Jordan, M.I., and Jacobs, R.A., "Hierarchical Mixtures of Experts and the EM Algorithm", *Neural Computation*, 6, p181-214, 1994.

[30] Leclerc, Y.G., "Constructing Simple Stable Descriptions for Image Partitioning", *Intl. J. of Computer Vision*, 3, p73-102, 1989.

[31] Lim, H.S., and Binford, T.O., "Structural Correspondence in Stereo Vision", *Proc. Image Understanding Workshop*, p794-808, 1988.

[32] Longuet-Higgins, H.C., and Prazdny, K., "The Interpretation of a Moving Retinal Image", *Phil. Trans. Royal Soc. London*, B208, p385-397, 1980.

[33] Longuet-Higgins, H.C., "A Computer Algorithm for Reconstructing a Scene from Two Projections ", *Nature*, 293, p133-135, 1981.

[34] Lucas, B.D. & Kanade, T., "An Iterative Image Registration Technique with an Application to Stereo Vision", *Proc. 7th Intl. J. Conf. on Artificial Intelligence*, pp674-679, 1981.

[35] Marr, D., and Poggio, T., "A Computation Theory of Human Stereo Vision ", *Phil. Trans. Royal Soc. London*, B204, p301-328, 1979.

[36] Marr, D., *Vision*, W.H. Freeman, New York, 1982.

[37] Matthies, L., Kanade, T., and Szeliski, R., "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences", *Intl. J. of Computer Vision*, 3, p209-236, 1989.

[38] Mellor, J.P., "Automatically Recovering Geometry and Texture from Large Sets of Calibrated Images ", Ph.D. Thesis. MIT Department of Electrical Engineering and Computer Science, Cambridge, MA., 1999.

[39] Papoulis, A., "Probability, Random Variables, and Stochiastic Processes", 3ed. McGraw-Hill: New York, NY, 1991.

[40] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., *Numerical Recipes in C*, 2ed, Cambridge University Press, Cambridge, UK, 1992.

[41] Rabiner, L., and Juang B.H., *Fundamentals of Speech Recognition*, Prentice Hall, New Jersey, 1993.

[42] Redner, R.A., and Walker, H.F., "Mixture Densities, Maximum Likelihood and the EM Algorithm", *Society for Industrial and Applied Mathematics Review*, 26:2, April 1984.

[43] Stenstrom, J.R., "Constructing Object Models from Multiple Images", *Intl. J. of Computer Vision*, 9, p185-212, 1992.

[44] Strang, G., *Linear Algebra and Its Applications*, Harcourt Brace Jovanovich Publishers: San Diego, CA, 1988.

[45] Szeliski, R., and Kang, S.B., "Recovering 3D Shape and Motion from Image Streams using Non-Linear Least Squares", *Journal of Visual Communication and Image Representation*, 5(1), p10-28, 1994.

[46] Teller, S., "Automated Urban Model Acquisition: Project Rationale and Status", *Proc. Image Understanding Workshop*, p455-462, 1998.

[47] Terzopoulos, D., "Multilevel Computational Processes for Visual Surface Reconstruction", *Computer Vision, Graphics, Image Processing*, 24, p52-96, 1983.

[48] Venkateswar, V., and Chellappa, R., "Hierarchical Stereo and Motion Correspondence Using Feature Groupings", *Intl. Journal of Computer Vision*, 15, p245-269, 1995.

[49] Wells, W.M. III, "Visual Estimation of 3-D Line Segments from Motion – a Mobile Robot Vision System", *IEEE Transactions on Robotics and Automation*, 5, p820-825, 1989.

[50] Wells, W.M. III, "Statistical Approaches to Feature-Based Object Recognition", *Intl. J. of Computer Vision*, 21, p63-98, 1997.